AIR FORCE HUMAN RESOURCES LABORATORY

ADVANCED ON-THE-JOB TRAINING SYSTEM: TRANSITION PLAN

John J. O'Connor

Ball Systems Engineering Division
2901 Juan Tabo NE
Suite 235
Albuquerque, NM 87112

TRAINING SYSTEMS DIVISION
Brooks Air Force Base, Texas 78235-5601

May 1990
Interim Technical Paper for Period August 1985 - December 1989

Approved for public release; distribution is unlimited.

AIR FORCE SYSTEMS COMMAND
BROOKS AIR FORCE BASE, TEXAS 78235-5601

NOTICE

This paper has been reviewed and is approved for publication.

HENDRICK W. RUCK, Technical Advisor
Training Systems Division

RODGER D. BALLENTINE, Colonel, USAF
Chief, Training Systems Division

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	May 1990	Interim - August 1985 to December 1989

4. TITLE AND SUBTITLE

Advanced On-the-job Training System:
Transition Plan

5. FUNDING NUMBERS

C - F33615-84-C-0070
PE - 63227F
PR - 2557
TA - 00
WU - 03

6. AUTHOR(S)

John J. O'Connor

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Ball Systems Engineering Division
2901 Juan Tabo NE
Suite 235
Albuquerque, NM 87112

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Training Systems Division
Air Force Human Resources Laboratory
Brooks Air Force Base, Texas 78235-5601

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

AFHRL-TP-89-98

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The purpose of the Advanced On-the-job Training System (AOTS) Transition Plan is to develop and document orderly plans for the transition of AOTS to the next stage of its development and for the movement of the operational system produced, from the prototype use to Air Force-wide deployment. This paper documents a technical analysis and assessment of the factors that must be considered in the transition of the program from an AFHRL technology demonstration program to an Air Force Automated Data System (ADS). The Transition Plan describes the prototype system used to establish a base from which the Transition Plan will depart, the steps and points necessary to progress to an operational system, and the recommended approach for the next stage of development.

14. SUBJECT TERMS

advanced on-the-job training system
expansion plan

15. NUMBER OF PAGES

155

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

AFHRL Technical Paper 89-98

May 1990

ADVANCED ON-THE-JOB TRAINING SYSTEM: TRANSITION PLAN

John J. O'Connor

Ball Systems Engineering Division
2901 Juan Tabo NE
Suite 235
Albuquerque, NM 87112

TRAINING SYSTEMS DIVISION
Brooks Air Force Base, Texas 78235-5601

Reviewed and submitted for publication by

Jack L. Blackhurst, Major, USAF
Chief, Advanced On-the-job Training System Program

This publication is primarily a working paper. It is published solely to document work performed.

SUMMARY

The Advanced On-the-job Training System (AOTS) was an Air Staff directed, AFHRL developed, prototype system which designed, developed, and tested a proof-of-concept prototype AOTS within the operational environment of selected work centers at Bergstrom AFB, Texas, and Ellington ANGB, Texas, from August 1985 through 31 July 1989. The purpose of the AOTS Transition Plan is to develop and document orderly plans for the transition of AOTS to the next stage of its development and for the movement of the operational system produced, from prototype use to Air Force-wide implementation. This paper documents a technical analysis and assessment of the factors that must be considered in the transition of the AOTS program from an AFHRL technology demonstration program to an Air Force Automated Data System (ADS). The AOTS Transition Plan provides for the movement of AOTS from the prototype proof-of-concept development stage to an operational system as either a Full Scale Development or Technology Insertion Program. The Transition Plan describes the prototype system used to establish a base from which the Transition Plan will depart, the steps and points necessary to progress to an operational system, and the recommended approach for progression to the next stage of development.

PREFACE

This paper was developed by Ball Systems Engineering Division, the AOTS on-site integration and management contractor, under Government Contract Number F33615-C-84-0070. The AFHRL Work Unit Number for the project is 2557-00-03. The primary office of responsibility for management of the work unit is the Air Force Human Resources Laboratory, Training Systems Division, and the Air Force AOTS manager is Major Jack Blackhurst.

FOREWORD

This technical report, D-R-005-89-34, Advanced On-The-Job Training System (AOTS) Transition Plan, is submitted by Ball Systems Engineering Division (BSED), Bergstrom Air Force Base, Texas 78743-5000, under the Integration and Site Management (I&SM) Program, prime contract F33615-84-C-0070, Contract Deliverable Requirements List (CDRL) item number 13.

The Air Force Human Resources Laboratory (AFHRL) has tasked the AOTS I&SM team to support transition and expansion planning. The purpose of the Transition and Expansion Planning task is to develop orderly plans for the transition of AOTS to the next stage of its development and for the movement of the operational system produced from prototype use to Air Force wide implementation.

This report documents a technical analysis and assessment of the factors that must be considered in the transition of the AOTS program from an AFHRL technology demonstration program to an Air Force Automated Data System (ADS). This analysis is a subtask under the Transition and Expansion Planning task and documents the current prototype, identifies the transition criteria and outlines the transition strategy to move to the next stage of development.

This document was prepared by principal investigators Mr. Darrel R. Knutson, Mr. Jerry Carlquist and Ms. Marianne Niven of BSED and Ms. Cynthia C. Curtis, Mr. G. Ray Sims, and Mr. Dennis R. Becker of the BDM Corporation.

Reviewed and approved by:

John J. O'Connor
I&SM AOTS Program Manager

TABLE OF CONTENTS

LIST OF FIGURES

EXECUTIVE SUMMARY

This Transition Plan (TP) provides for the movement of AOTS from the prototype proof-of-concept development stage to an operational system as either a Full Scale Development (FSD) or Technology Insertion Program. The first portion of this plan describes the prototype system to establish a base from which the TP will depart. The next portion describes the steps and points necessary to consider in progressing to an operational system. Finally, the TP addresses the recommended approach to progress into the next stage of development.

A. Background. The current On-the-Job Training (OJT) system is "paper intensive", cumbersome, and has a large "hidden cost" that is difficult to quantify because it is buried in the day to day operations necessary for mission accomplishment. However, advanced technology can be brought to bear on the problem and a better, more fully useful system produced. At the same time, the excellence and standardization of training can be increased Air Force wide while personalizing the training to the individual's requirements.

1. Operational Need. The AOTS technology program was started in the early 1980s in response to operational requirements for improving On-the-Job Training (OJT). Serious deficiencies in the OJT system were first addressed in an Air Force Human Resources Laboratory (HRL) study in the early 1970s and documented by the Air Force Inspector General (IG) in an April 1977 Functional Management Inspection (FMI) report. While individual Major Commands (MAJCOMs) attempted to resolve many of these problems, the scope, importance, and diversity of the Air Force OJT requirement did not lend itself to individual MAJCOM solution and the underlying problems remain today. Additionally, the magnitude of the OJT requirement has increased during the ensuing years because of decreased funding for Air Training Command (ATC) sponsored formal training courses and increased sophistication of the newer weapon systems. Today, an estimated 70% of all technical training requirements are met through OJT. Beyond the significant operational need for an improved OJT is the impact on the morale of Air Force personnel. The existing system, with its ad hoc approach and wide variation of effectiveness has a large impact on job satisfaction and the ability of the Air Force to compete with the civilian job market for qualified, skilled personnel.

2. Payoffs. The Advanced On-the-Job Training System (AOTS) will produce many measurable and immeasurable benefits for the Air Force. Through the use of computer technology, the immense paperwork burden associated with the current training administration and documentation can be significantly reduced. At the same

time, the quality of the training available can be standardized at a higher level of excellence by using state-of-the-art training technology. This is possible because advanced technology will allow the development of highly detailed Computer Aided Instruction (CAI) modules developed by knowledgeable, well qualified instructors that can be presented without requiring a teacher to be present. This is in contrast to the present ad hoc approach to training where availability of an instructor can be an obstacle. The addition of an ability to immediately aggregate the level of training, training requirements and other training factors at the supervisor, commander, base, command or Air Force level provides a significant nearly real time force management capability that is presently not available. This ability to identify and update job performance and mission training requirements will allow the concentration of training in the areas of greatest concern and relevance.

B. Road Map. This is a technology introduction program designed to show the application of computers and state-of-the-art training technology to solve a real Air Force wide problem. There are two phases to this program: prototype development, and operational production and implementation.

1. Prototype Development. In this stage, the prototype AOTS was developed as a proof-of-concept. This was accomplished in the program described in Section I. The prototype was developed using a main frame architecture and dedicated equipment and communications. Some of the many paybacks from this were the development of requirements for a follow-on system incorporating Air Force standards and Standard Systems and the development of a baseline of OJT costs.

2. System Development and Implementation. AOTS can be developed to a production/operational system using one of several different architectures, under a Full Scale Development or Technology Insertion approach and implemented in a number of ways. The factors affecting the implementation of the AOTS in the Air Force are difficult to separate from the factors affecting the production decisions and these two major areas are discussed together. The decisions necessary to proceed to the production stage are discussed in Section II. The decision process necessary to establish the production system is discussed in Section III.

3. Recommended Alternative. The AOTS program can solve a significant Air Force problem. The prototype has proven the concept that technology can provide real solutions to important OJT deficiencies. The prototype program showed that the best solution would be to use a Full Scale Development program to

produce a networked personal computer based Advanced On-the-Job Training System and implement it Air Force wide.

SECTION I - TECHNOLOGY CAPABILITY DESCRIPTION

This section is a description of the underlying technologies of the AOTS prototype program -- How the program was managed and who participated, and what the final products are. For a full definition of how the system is to be used, see the AOTS Operational Concept Document (OCD) and Operational Guide.

A. Objective. The AOTS technology program was started in the early 1980s in response to operational requirements for improving On-the-Job Training (OJT). Serious deficiencies in the OJT system were first identified by HRL in the early 1970s and documented by the Air Force (AF) Inspector General (IG) in an April 1977 Functional Management Inspection (FMI) report. While individual Major Commands (MAJCOMS) attempted to resolve many of these problems, the scope, importance, and diversity of the Air Force OJT requirement did not lend itself to individual MAJCOM solution and the underlying problems remain today. Additionally, the magnitude of the OJT requirement has increased during the ensuing years because of decreased funding for Air Training Command (ATC) sponsored formal training courses and increased sophistication of the newer weapon systems. Today, an estimated 70% of all technical training requirements are met through OJT.

1. Hidden Costs. The cost for this steadily increasing workload is hidden and not necessarily directly quantifiable. The Air Force continues to accomplish its mission even though OJT requirements have become an ever increasing burden to both supervisors and airmen. They are spending increasingly more time in training and training documentation rather than on direct mission support. No additional manpower authorizations have been (or are likely to be) made due to this situation. This ever increasing paperwork burden for the work center supervisor has reduced the time available to provide quality training while at the same time, training responsibilities have significantly increased.

2. Technology Program. The AOTS technology program was established as a first step towards creating an improved OJT system that is usable throughout the Air Force. The goal was to demonstrate that computer technology could be used to improve the system by:

(a) Reducing the paper work burden associated with administering and documenting OJT to the more than 300,000 airmen in over 300 specialties participating in OJT upgrade/qualification at any one time.

(b) Improving and standardizing training delivery and evaluation.

(c) Providing better identification and updating of job performance and training requirements relevant to the mission.

3. Prototype. To accomplish these objectives, the Air Force Human Resources Laboratory (AFHRL) embarked on the AOTS technology demonstration program that included developing a prototype system which integrated several underlying computer based capabilities into a functional training system for four Air Force Specialties (AFSs). To evaluate the effectiveness of the prototype in meeting the above stated objectives, a one year System Level Test and Evaluation (SLT&E) activity was made an integral portion of the overall demonstration effort. The prototype development effort was to reuse already developed government owned software to the maximum extent possible and provide:

(a) Computer based documentation and tracking of individual training permitting supervisors and trainees to concentrate on the training and not on the administrative paper work burden associated with manual documentation of the training.

(b) Computer assisted matching of operational job specific performance objectives, associated training requirements, and evaluation criteria to ensure that job-site training is producing quality airmen fully qualified to exercise the skills necessary to perform a pre-determined set of tasks related to specific duty positions.

(c) Computer tracking of job task certification and monitoring of certification to provide for a review of task qualification within a user specified time period.

(d) Computer aided development of multi-media instruction modules that incorporate state-of-the-art text manipulation, color graphics, and interactive video displays with a reprogrammable alternative selection capability that can be upgraded or modified easily and quickly.

(e) Computer based delivery of multi-media training modules combined with real-time evaluation and immediate user feedback to determine job knowledge deficiencies and prescribe the remedial training required to achieve the minimum standards of the specific duty position.

4. Prototype Limitations. The prototype program was a technology effort and was limited to proving the concept. It was not intended to represent an operational or production system. The software function and supportability features implemented were limited to the minimum necessary to achieve this primary proof-of-

concept purpose. Likewise, the hardware architecture used only represented the vehicle to implement the software. A secondary purpose of the prototype was to document the changes, functions and architecture needed for a production system. Since the prototype had and achieved these specific purposes, it would be risky to try to directly implement it as an Air Force operational system.

B. Approach. The primary mission of AOTS is to train airmen with the necessary job skill qualifications in operational duty positions quicker and at lower cost than the existing cumbersome, manpower intensive OJT system. The development and test program, therefore, was structured to provide the maximum possible operational influence and feedback on the design to the Air Force for evaluating the effectiveness of the prototype and establishing optimum criteria for a follow-on Full Scale Development (FSD) program. This lead to implementing the prototype at operational bases and executing SLT&E in the operational environment. The technology program, however, was limited in cost and scope. This meant only a few AFSs and a few bases could be represented in the development and test efforts. Additionally, it made sense to maximize the reuse of government owned software such as the Instructional Support System (ISS), currently supported by AFHRL, which was developed to provide computer aided courseware development and delivery.

1. AOTS Program Management. AFHRL established the AOTS technology program office at Bergstrom Air Force Base (AFB), TX in 1985. McDonnell Douglas Corporation's Douglas Aircraft Company (DAC) was competitively selected to design, develop, test and evaluate the prototype AOTS. The Ball Systems Engineering Division (BSED)/BDM team was competitively selected to provide engineering support to the program office. Program office manning included contractor and Air Force personnel who were experts in training and Air Force personnel who were Subject Matter Experts (SME) in the demonstration AFSs.

(a) The AOTS technology demonstration program was a three phase effort scheduled over a four year period. During the first phase, preliminary design of the AOTS prototype was completed. Phase II consisted of the detailed design and development activities. Finally, implementation and SLT&E at Bergstrom Air Force Base (AFB) and Ellington Air National Guard Base (ANGB), TX were scheduled for accomplishment during Phase III. Figure 2 on Page 12 depicts the AOTS program schedule.

(b) Installation of AOTS was completed at Bergstrom AFB in August 1988 and at Ellington ANGB in September 1988. SLT&E is in progress and should be completed by 31 July 1989. Results are expected to be available by 1 October 1989.

2. Prototype. The prototype system was developed and installed in work centers at Bergstrom AFB and Ellington ANGB. The prototype software was hosted on an AFHRL owned Digital Equipment Corporation VAX 8650 mainframe computer installed at Brooks AFB, TX. work center terminals (Z-248s) and associated peripherals were connected to the VAX through fiber optics communication lines. All processing was done centrally on the VAX mainframe.

3. Air Force Specialties. AFSs that were selected for inclusion in the prototype AOTS are: Jet Engine Mechanic, 426X2 (now 454XO/A); Aircraft Maintenance, 431X1 (now 452XD/M); Personnel, 732X0; and Security Police, 811XX. Together these AFSs represent approximately 20% of the Air Force work force, and include diverse job skills and training requirements. All Air Force components (Active, Reserve, and Air National Guard) participated in the demonstration and test effort. work centers for the reserves and active Air Force were located at Bergstrom AFB, TX. National Guard work centers were located at Ellington ANGB, TX.

4. Test & Evaluation. A formal test program consisting of two distinct activities was used. The first activity was the development activity and was primarily directed towards proving the design and providing feedback to designers to allow improvements during the development stage. The second activity, System Level Test and Evaluation (SLT&E), primarily addressed suitability (Does the AOTS overcome identified deficiencies and can it be used throughout the AF?) and acceptability (Is AOTS accepted by the users as friendly and easy to use?) of the AOTS. Issues such as compliance with the specifications and performance were addressed during both activities.

(a) Planning for SLT&E was strongly influenced by a variety of practical constraints. One major factor was the statistical limitations imposed by the limited number of AFSs and bases. The use of "control" groups was limited by the existence of single offices in some of the test AFSs. This also limited the sample size. Additionally, because AOTS was tested in an operational environment, no actions could be taken that would hinder the day-to-day mission of the work centers. Despite these constraints, the advantages of testing the system in the "real world" outweighed the disadvantages, particularly in the areas of suitability and acceptability.

(b) SLT&E commenced 1 August 1988. An interim report was produced in May 1989 and the final report is due 1 October 1989.

C. Payoffs. The AOTS technology program was structured to provide several mutually supportive benefits towards the ultimate goal of maintaining maximum combat readiness by providing highly qualified and skilled technicians. The prototype was targeted to demonstrate that computer technology could be used to significantly enhance the OJT process from the aspects of training management, course development, course delivery and skill evaluation. It accomplished this objective. Results from the SLT&E are not yet fully documented, but initial reports indicate that supervisors and trainees both use and like the system. Beyond this proof of concept, the program provided several ancillary products that can be used by the Air Force to improve OJT.

1. Design Specification. Adequate data was gathered to permit the development of system specifications that can be blended with user defined requirements to determine follow-on program goals. This provides significant risk reduction for follow-on development efforts whether they are executed as a classic FSD program, or the AOTS technologies are inserted into other ongoing programs such as the Core Automated Maintenance System (CAMS), Personnel Concepts III (PC III), and Security Police Automated System (SPAS).

2. OJT Costs. Program evaluation tools were established to assess OJT costs and overall OJT program effectiveness. This determination of actual costs had been a most difficult problem to identify. After all, OJT appears to be "free." It is done on the job during "spare" time and does not require dedicated resources. Time spent in OJT is not documented. The costs attributable to a technician's inefficiency which are exclusively caused by a lack of training are extremely difficult to quantify. Further association of those costs with the cumbersome OJT process and measuring improvements resulting from an AOTS is even more difficult.

D. Prototype Description.

1. Hardware system. The AOTS prototype is a centralized software package operating on a VAX 8650 using Zenith 248s as remote terminals. The mainframe computer is located at Brooks AFB, TX and linked by fiber optics to 26 user work stations at Bergstrom AFB and nine work stations at Ellington ANGB. An additional 48 developmental work stations were installed at Bergstrom AFB. All the peripheral devices are separately controlled by the mainframe computer. This includes the printers, scanners, and other devices used by the work stations. A generalized view is shown in Figure 1 - Prototype Hardware Architecture.

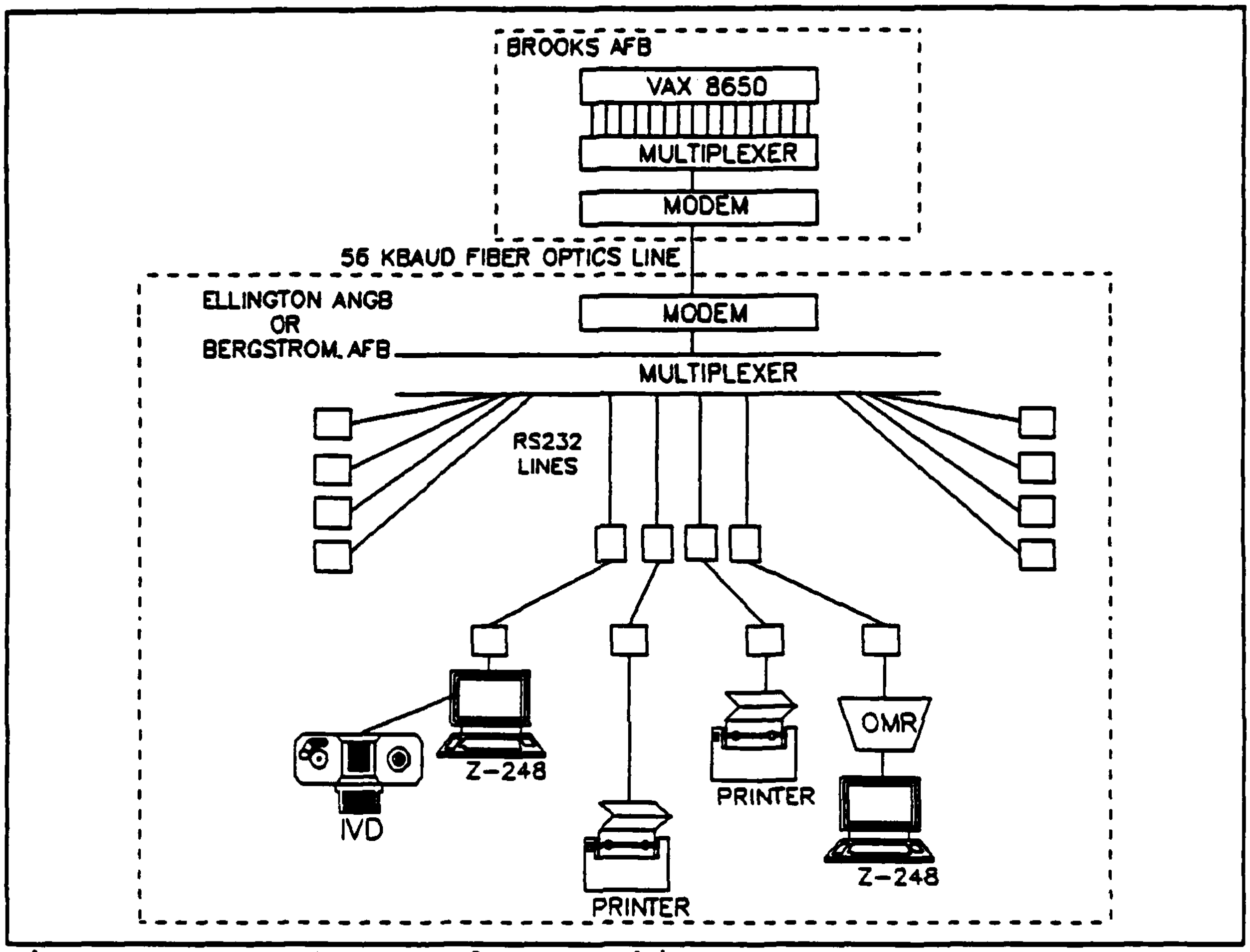

Figure 1. Prototype Hardware Architecture

2. Top Level Software Description. The AOTS software package is primarily an Ada language program but also includes FORTRAN, ASSEMBLER, and Statistical Analysis System (SAS) language modules. See Appendix A for a complete breakdown. The overall program is designed to provide three functional capabilities which are: 1) management, 2) evaluation, and 3) training development and delivery. A fourth module provides necessary access to computer support tools.

(a) Management Subsystem. This subsystem supports the management of training requirements, managing the airman's actual training received and scheduling training. The training requirements management function supports the actual development and maintenance of the Master Task Lists (MTL) of job related tasks and the Other Training Requirements (OTR) list as well as maintaining task performance and proficiency data. The airman training management function allows the generation of an airman's training record, the diagnoses of training needs and the monitoring of progress.

(b) Evaluation Subsystem. This subsystem allows the management of evaluation instruments, the actual evaluation of task training, and the application of quality control. The management of evaluation instruments allows the developer to plan, develop and deliver evaluations using a test item bank. In the evaluate performance area, the trainer actually evaluates and records task and knowledge proficiency data. The quality control function allows the overall assessment of the training program and the generation of system reports.

(c) Training Development and Delivery Subsystem. This subsystem provides for the development and delivery of training primarily using Computer Aided Instruction (CAI) modules. This function is primarily provided through the separately developed and government furnished Instructional Support System (ISS). The training development function provides the training developer with the mechanism to interactively work with the ISS to produce and maintain CAI modules. In the training delivery function the trainee has the ability to access the stored instructional material and CAI and receive on-line training. The subsystem can also deliver interactive video disk (IVD) lessons.

(d) Computer Support Subsystem. This subsystem consists of the software necessary to make the system function. The software described in paragraphs D.2.(a), (b), and (c) above are the functions of AOTS. The Ada and FORTRAN compilers, operating system, configuration management tools and documentation as examples are all controlled under this subsystem.

3. Required Improvements. The prototype was designed to provide a proof-of-concept, not incorporate all functions of an operational system. The follow-on to this prototype must incorporate additional or expand some existing functions to provide full operational capability.

(a) The single most important capability that must be added is the ability to incorporate new or different AFSs. The current four AFSs are "hard coded" into the prototype and changing or adding new AFSs involves some rewrite.

(b) Another important function that needs to be incorporated is the ability to electronically communicate with other systems such as personnel (Personnel Concepts III (PC III)) and Functional Area management systems such as the Security Police Automated System (SPAS).

(c) The follow-on system should include the ability to allocate, schedule, account for, and report on the use of OJT resources. This would include the ability to record training time by module, activity, etc. as well as the time spent on individual CAI lessons or specific tasks by module or lesson. Also necessary would be the inventory management and scheduling of physical resources such as classrooms or training devices. An improved user interface throughout the system should also be considered.

(d) Supportability of the system should also be improved. Here there are several areas that may require more design effort prior to implementation. Chief among these areas are the need for better documentation, configuration management and program management tools. See section II. E. for more information.

(e) The lessons learned in the prototype development will be documented at the end of the SLT&E and should be examined for follow-on changes needed.

E. Data and Documentation Deliverables. During the course of the AOTS program, contractors and government sources have produced many documentation items. A listing of these documents along with their producer or office of primary responsibility (OPR) and date of delivery/publication are available in Appendix D. Any dates listed in parenthesis indicate submittal dates beyond the date of this report. All documents listed are part of the AOTS technical library and will be turned over to the follow-on development agency or archived for future use upon program completion.

F. Schedule. The AOTS program was initiated with a Request for Proposal (RFP) and Statement of Work (SOW) initiated in July 1983 under the authority of the Air Force Human Systems Division (HSD). Concurrent with the development program was the initiation of a contract for Integration and Site Management (I&SM). The I&SM contract was competitively awarded to the team of VERAC, doing business as Ball Systems Engineering Division (BSED) and the BDM Corporation, starting on 15 March 1985 and the prototype development was awarded to Douglas Aircraft Company (DAC) starting on 1 August 1985. A 48 month, three phase schedule including a one year System Level Test and Evaluation (SLT&E) was proposed. The original Phase I Design slipped two months but, the overall 48 month schedule has been firmly adhered to. The only modification to the schedule was the extension of the I&SM contract to coincide with the development contract. The overall schedule is depicted at Figure 2.

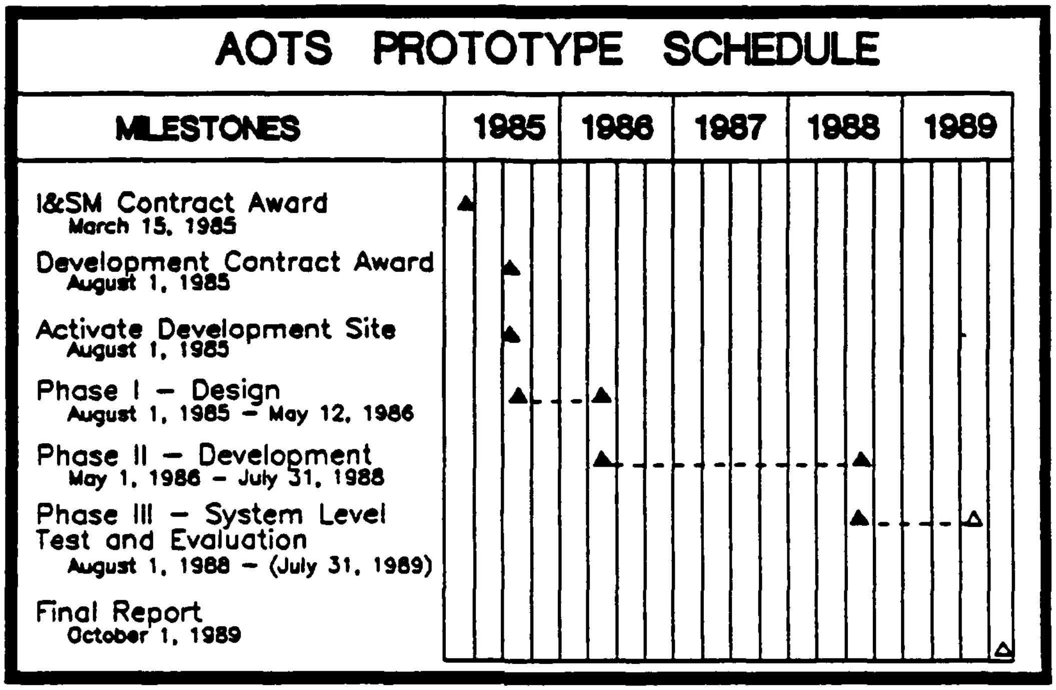

Figure 2. AOTS Prototype Schedule

G. Funding. The AOTS program is a Research and Development (R&D) effort included in the HSD budget. It has been entirely funded by the Human Resources Laboratory under the auspices of their tasking to improve the overall training level at a reduced cost. Total cost of the prototype program when complete will be $11.7 Million.

SECTION II - TRANSITION CRITERIA

This section discusses the requirements, effort and influencing factors in moving from the prototype described in Section I to a production system. Appendix A fully describes the prototype AOTS system and Appendix B describes the changes and evolution necessary to develop a production system. The two possible methods of implementing AOTS are: 1) a full scale development (FSD) program managed by a system program office (SPO) and 2) the incorporation of selected portions of the prototype design into other standard Air Force systems.

This plan is primarily designed to describe the factors to consider if a FSD program is established. However, many portions of the plan are also applicable if AOTS is implemented as a technology insertion activity through other standard systems. In that case, the SPO for the other system would be responsible for incorporating the factors discussed in paragraphs A through F below.

A significant portion of the AOTS technology demonstration program activities were focused on establishing requirements for the production AOTS. Preliminary results from the SLT&E provided measurements of the prototype's suitability and acceptability. Lessons learned, and user inputs, were used to establish a tentative set of performance requirements. Three possible architectures were then evaluated against each other: 1) mainframe serving one or more bases; this is essentially an extension of the prototype architecture, 2) standalone personal computers, and 3) personal computers networked to a centrally located file server mini-computer at the base level.

All three architectures are considered in the discussions below. Specific efforts or influencing factors are pointed out where applicable.

A. Performance Measurements. There are numerous possible measures that can be applied to characterize a system. Fundamental to establishing these measures is a clear understanding and statement of the decision criteria for evaluating these measurements. The ultimate goal of an AOTS is to provide better trained personnel through a more standardized OJT, faster, with less effort, and at less cost than can be realized through the existing system. A secondary goal of the prototype program is to establish measurements that permit evaluation of AOTS' ability to meet that ultimate goal. These measurements are defined through an iterative process. The first step is to state meaningful performance objectives at the general level. For example, AOTS must be sufficiently user friendly to encourage supervisors, trainers, and trainees to readily use the system. The second step is to "interpret" those generalized performance goals into performance measurements. For example, the average length of time required for supervisors, trainers, and trainees to learn to effectively use the system. The final step is to define the system's physical attributes that directly contribute to the performance measure and establish "quantitative" measures of these characteristics. For example, throughput, time lag between user input and computer response, tolerance to input errors from the operator, and reliability can be measured in units of time.

Typically, SLT&E results and lessons learned are the best sources for establishing which of the many possible performance measures are most valid for describing a system's ability to meet its overall goal. These are not yet available for AOTS, but can be used for that purpose when they are collated and delivered in October 1989.

B. Quantitative Measurements. Quantitative measurements are the physical measurements which support objective evaluation of the system's ability to meet desired performance criteria. Again, using a process similar to that for determining valid performance measures, quantitative measures are developed by establishing physical criteria that contribute to a performance measure and then evaluating their individual importance and ranges of tolerance. In the example above, throughput may not be as important a measure as tolerance to operator error. Or there may be a threshold beyond which decreases to the time lag between user input and system response do not increase user friendliness.

Performance measures are the source for establishing these physical criteria. SLT&E results can then be used to evaluate their relative importance through multiple sensitivity analyses. These activities should be completed and criteria established as soon as possible after the test results are available but not later than prior to establishing development specifications for a follow-on FSD program.

C. Producibility. The prototype AOTS system, described in detail in Appendix A, was developed using a centralized mainframe computer architecture. This architecture is appropriate for a prototype system; however, a production system is dependent on user requirements and is likely to be required to conform to other system architectures, and to run on other hardware and operating systems. There are three primary architectures under consideration for the FSD AOTS. They are the centralized mainframe based architecture, a standalone personal computer (PC) based architecture, and a networked PC architecture. The paragraphs below describe each of these architectures, and discuss the changes in the prototype AOTS software required to produce an FSD AOTS conforming to each architecture. Some changes will probably be required no matter what architecture is selected for implementing an FSD AOTS.

1. General Changes Required. Depending on actual user defined requirements, several functional enhancements should be made regardless of the architecture used in the final implementation. These enhancements fall into three categories: 1) changes to generalize the software to support more than or change the 4 prototype AFSs, 2) changes to support interfaces to other Air Force systems, and 3) other changes. These changes are outlined below and the impact of the changes are discussed in the appropriate paragraph.

(a) The prototype AOTS supports only four AFSs. An FSD AOTS must be capable of supporting any of the 250-400 AFSs used by the Air Force. Even if a user only wanted four AFSs, it is unlikely they would want the same four in the prototype. The four test AFSs are hard coded in the prototype software menus and routines which access, manipulate, and store data and produce reports. Since the primary function of the AOTS is data management, this means that these four AFSs are hard coded into a percentage of the prototype software packages.

(b) The AOTS prototype includes data which originates from or is also used by a number of other Air Force Systems. For the purposes of the prototype, data which originated from other systems was loaded into AOTS via magnetic tape. In an FSD AOTS system, electronic interfaces to exchange data with other Air Force Systems would be necessary to provide a more efficient and timely means of sharing data and ensure that both AOTS and the other systems contain up-to-date information. Systems considered for possible interface with AOTS include PC III, CAMS, SPAS, and the Advanced Training System (ATS).

(c) Because the AOTS prototype implemented a fairly complete set of functions, the prototype AOTS software includes nearly all the functionality required for an FSD system. The only function which was omitted from the prototype, which appears to be essential to a production system, is the ability to automatically backup the system, data bases or records and to restore them when necessary. Another change which could be required to convert the prototype software into a FSD version would be to increase any limits on the size of the AOTS databases to meet reasonable size requirements of an FSD system and increase the supportability of the software. Useful, but not essential, enhancements would add the training resource and inventory management functions and a training resource use/accounting capability. The scope and magnitude of the effort to incorporate these enhancements into AOTS has not been accomplished and would need to be analyzed prior to implementation. All changes in this category are either of small magnitude or not clearly necessary for an FSD AOTS. Therefore, the effort required to accomplish these "other changes" was not estimated.

2. Centralized Mainframe Based Architecture. In a centralized mainframe architecture the software for the system runs on a single, central computer. Multiple users may gain access to this software simultaneously through terminal devices connected to the central computer. Peripheral equipment, such as printers and scanners are controlled from the central computer, although they may be connected to the computer through terminal devices. The data which is stored, retrieved, and manipulated by the software also resides on storage devices connected to the central computer, rather than being distributed across individual terminal devices, under this architecture. Strengths of a centralized architecture include simplified software design, and good control over software configuration. Weaknesses of this architecture include relatively low reliability (if the main computer goes down, the whole system is down), inability to support deployment exercises (mainframes are not portable), and potentially high hardware cost for locations with small numbers of users.

(a) The prototype AOTS software was developed on a VAX 8650, using Zenith Z-248 personal computers as dumb terminals. Under the prototype architecture, all AOTS software and data reside on the VAX and, while they could be downloaded for deployments, would require a similar VAX system at the deployment site. Control of peripheral equipment, including printers and scanners, is also accomplished from the VAX. The only software which runs on the Z-248s under the prototype AOTS architecture is a terminal emulator package. The changes required to convert the prototype AOTS into an FSD system using the centralized mainframe architecture are therefore limited to software enhancements necessary to support FSD requirements.

(b) The analysis detailed in Appendix B concludes that approximately 1,800 lines of code, or about 1% of the AOTS prototype software would have to be rewritten to support more than the four AFSs in the prototype system. The effort required to add the desired automatic interfaces to AOTS is approximately 3000 lines of code or about a 1.5% change.

(c) If the prototype AOTS is both ported to a new host computer and changed as discussed in paragraphs C.1.(a)-(c) above, then 62,000 lines of source code, or 29% of the AOTS prototype software would have to be modified or rewritten.

3. Standalone Personal Computer Architecture. In a standalone personal computer based architecture, the complete AOTS software would be ported to (converted to work on) a personal computer. The software would run entirely on the PC microprocessor, and would support a single user at a time. If more than one user needed access to the AOTS software simultaneously, a second PC running a second copy of the AOTS software would be required. Data used by AOTS, such as Master Task Lists (MTLs), Generic Position Task Requirements (GPTRs), or Airman Training Records (ATRs) would be stored locally on the PC. New data could be loaded into the PC via floppy disks, tapes, or by modem from a remote system. The strengths of a standalone PC architecture include portability for deployment, high reliability in comparison with the central mainframe architecture, and low hardware cost for small installations. This architecture has two primary weaknesses: 1) the relative difficulty of distributing software updates and providing users with current data (e.g., MTLs, GPTRs, courseware) and 2) the difficulty of collecting summary data for use on management evaluation of OJT and the AOTS itself.

(a) In order to convert the prototype AOTS software to operate in a standalone PC environment, the software would have to be ported to a new hardware and operating system environment and modified to operate as a single-user rather than a multi-user system. This conversion is discussed in more detail in Appendix B. In order to accomplish this conversion, an estimated 48,000 source lines of code, or approximately 24% of the prototype software would have to be modified or rewritten.

(b) This 24% rewrite includes the AFS related changes but does not include interfaces to other systems. If the effort required to implement interfaces to other systems is added to the effort required to convert the prototype software to a standalone personal computer architecture, the total change estimate is a 26% rewrite of the prototype software.

4. Networked System Architecture. In a networked system architecture, multiple copies of the AOTS software would run on personal computers located in the work centers where OJT is accomplished. These personal computers would be connected via a local area network to a minicomputer or super-microcomputer which acts as an AOTS file server. Data such as MTLs, GPTRs, OPTRs, ATRs, and courseware would be stored at the file server, and made available to all PCs as needed. The file server software would also provide the capability to connect to the base DDN gateway. Since the AOTS software would run locally on the PCs, individual PCs could be disconnected from the network and taken along on deployment exercises. Any data needed while deployed would be

copied onto floppy disks or other portable media from the file server prior to deployment. This architecture combines the strengths of the central mainframe and standalone PC based architectures, and is recommended as the target architecture for an FSD AOTS.

(a) In order to convert the prototype AOTS software to operate in the networked system architecture, the software would have to be ported to a new hardware/operating system environment, modified to support a single user at a time (per copy of the software), and rewritten to support remote storage/retrieval of AOTS data. As described in Appendix B, converting the prototype AOTS software to the Networked System architecture would require changing an estimated 88,000 lines of source code, or approximately 36% of the prototype software.

(b) This estimate does not include the effort required to add interfaces to other systems to AOTS. Adding the interfaces and effort required to convert the prototype software to the networked architecture results in a total estimated modification/rewrite of 91,000 lines of source code, or 38% of the prototype.

5. Recommended Architecture. Despite the slightly higher percentage of change required for a networked architecture, its flexibility in supporting both normal and deployed operation, higher reliability, and the ease with which software and data updates can be accomplished, the networked system appears to be the best architecture for an FSD AOTS.

(a) There are three possible approaches to producing AOTS in this architecture: 1) convert the prototype software to operate in a networked architecture, 2) convert the prototype software to the standalone PC architecture and then use Preplanned Product Improvements (P3I) to evolve into the networked system architecture, and 3) write the FSD AOTS software from the ground up, making use of the prototype AOTS design to simplify the development process and reduce development risk.

(b) As discussed previously, converting to a networked system environment would require a 36% rewrite of the AOTS prototype software, and converting to the standalone PC architecture requires a 24% rewrite (without taking into account the effort required to accomplish P3Is). The percent software rewrite required to convert the prototype software to either the PC or the Networked System architecture is high enough to question whether rewriting the AOTS software from scratch might be more life cycle cost effective than modifying the prototype software. A life cycle cost analysis comparing a complete rewrite (based on the prototype design and algorithms) with modifying the prototype software should be performed prior to deciding on the best approach to produce an FSD AOTS.

6. Implementation Method. Paragraphs C.5.(a) through C.5.(d) make the assumption that the FSD AOTS will be implemented as a new Air Force Automated Data System (ADS) standard system. This is the preferred method because it is the best way a completely standardized system can be implemented throughout the Air Force. Implementation by only some of the MAJCOMs would be the same but is not recommended because of the loss of standardization. Another possibility is that all or part of the AOTS functionality could be implemented as part of another standard system. Without knowing what subset of the AOTS functionality would be required in another system, or what the architecture of the other system is, it is impossible to estimate the effort required to implement AOTS in that manner.

D. Affordability. The trade-off among the costs incurred to implement a new system, the operational costs avoided by implementing the new system, and the potential benefits derived from the new system are common issues in all acquisition decisions. The affordability of any new system depends on the blend of all these factors. Any discussion on costs must first establish the factors bearing on the costs. In the case of AOTS, these general factors must translate to their impact on the cost of developing the tool. Then the cost of using the tool will lead to the benefits obtainable.

1. General. A PC-based cost estimation tool, the AOTS Life Cycle Cost Model (ALCCM), was developed to allow easy assessments of alternative deployment concepts, postulated benefits from an operational AOTS, and budget forecasting for program development. The ALCCM uses hardware costs from existing DoD "standard" contracts such as the Zenith Data Systems Z-248 personal computer contract and the AT&T AFCAC 251 standard contract (Standard Multiuser Small Computer Requirements Contract). The software development costs were estimated by using a parametric cost model, PRICE S, which is a robust, proprietary cost estimation tool. These PRICE S software development costs were compared to costs estimates using a public-domain spreadsheet version of the COnstructive COst MOdel (COCOMO) and the Air Force Systems Command (AFSC) Contract Management Division version of COCOMO called REVIC. The hardware and software development costs were combined in the ALCCM to project the costs of transitioning from the prototype AOTS to a production system, then supporting the AOTS for a 10 year life cycle.

(a) The ALCCM was developed using established cost estimating relationships (CERs) in a menu driven, PC-based relatively user-friendly environment (Lotus 1.2.3). This allowed the user to describe the hardware, software, courseware, and other operational parameters in simple terms, then calculate the associated development and annual operating costs automatically. The original AOTS cost model was reviewed by various Air Force agencies, including the Air Staff, and was deemed to provide a credible first order estimate of the life cycle costs of AOTS. The output can be viewed on an Air Force-wide basis, by MAJCOM, by 2-digit AFS, by major Air Force base location, or by combinations of the above. An additional output feature distributes the estimated costs over a user-supplied number of years and sums the costs in a form suitable for generating program budget submissions for the Program Objective Memorandum (POM) cycle (ie., 3600 [RDT&E], 3080 [Procurement], and 3400 [O&M] dollars).

(b) ALCCM allowed the performance of sensitivity analysis to determine the cost drivers for the system. The architecture selected for implementation was not a large factor because the predicted software development costs were heavily dependent on the delivered source lines of code (SLOC). From our detailed analysis, it appears that the prototype software will have to be modified from 24% to 36% depending on how the production version is implemented, and which operational features will be modified. Although different components of the software will need modification for the "main frame" vis-a-vis the "networked" version, the estimated amount of change remained within the 24-36% range.

2. Software Costing. The cost of the software development was based on modifying a percentage of AOTS SLOC to provide the full operational features, and to be "re-hosted" to another computing environment. For this analysis, the "other system" was the AT&T 3B2/600G system now available to government agencies via the AFCAC 251 standard contract. The re-hosting effort included a change from the prototypes's operating environment (DEC VAX 8650; VMS operating system) to the UNIX-based 3B2 environment. The estimated SLOCs were used as inputs into the three software development costing models mentioned above. Additional input parameters were used to represent the programming language, operating environment, management complexity, internal integration, fraction of storage space utilized, schedule constraints, personnel attributes (i.e., experience with the particular language, product familiarity, use of software tools), and degree of design stability. The costs generated by the PRICE S were used because HSD was most familiar this model and will rely on costs generated by PRICE S for future Major Automated Information System Review Council (MAISRC) processes. Essentially equivalent runs using the two

versions of COCOMO indicated that the PRICE S costs were reasonable estimates of an AOTS full scale production effort. A key driver to all of the software estimation models is the expected size of the software. "Size" here is a computed metric that is different in PRICE S and the two versions of COCOMO. The common factor is the estimated SLOC. "Source" lines are not the total lines of code one would see in a complete print out of the AOTS software since comment lines and blank lines are omitted from the count. The SLOC for the completed prototype AOTS software project ranged from a low of 195,720 (representing 29% redesign/rewrite) for the "stand alone" PC-based architecture to a high of 242,700 (36% redesign/-rewrite) for the "networked" system architecture.

(a) General Approach. The costs of ownership of an AOTS were estimated from two distinct perspectives:

(1) Case 1: included the costs of acquiring and supporting the advanced training capability demonstrated in the AOTS prototype.

(2) Case 2 included the Case 1 costs of developing and supporting an AOTS capability. Case 2 also included the cost of actually conducting job-site training for the same 10 years.

(b) Case 1: Cost of the "Tool". The costs of the Full Scale Development (FSD) required to convert the prototype AOTS software into a production version are primarily costs assigned to the developing MAJCOM, AFSC in our assumed scenario. The above software costs were combined with the associated development hardware, and communications hardware. These costs and other input factors were processed through the CERs in the ALCCM to generate the total costs of acquiring an AOTS capability. MAJCOMs other than AFSC were similarly "billed" for the costs of acquiring the AOTS capability, but without the software development costs of the FSD effort. In addition, there are other costs that, while not part of the AOTS FSD program, must be considered in the total cost to develop the capability. These include the commercial off-the--shelf (COTS) hardware required for system operation, the identification and development of the Computer Aided Instruction (CAI) or courseware and the cost of converting from the existing system to the new system. Operating MAJCOMs were assumed to budget for and incur the procurement costs of AOTS-specific equipment. In addition, each MAJCOM was assigned the costs of development of courseware and providing necessary computer-to-computer communications. In addition to the MAJCOM specific cost allocations, additional runs of the ALCCM produced Functional Area-specific cost projections. For example, the cost of acquiring an AOTS capability for all the Security Police (AFS 81) in the Air Force enlisted core was calculated. As another example, all of the specialties in the Logistics/Maintenance career fields were separately costed (AFSs 40 through 48). These costs are a part of the overall cost of AOTS but not the FSD program to produce the operational system.

(1) For the "main frame" architecture, approximately 29% of the SLOC required "redesign/rewrite" to develop an operational AOTS. Hence, the ALCCM used the PRICE S generated cost of $11.5M for a calculated 22 month FSD effort. A complete rewrite of the code using the "main frame" concept design was estimated by PRICE S at $19.1M, with an estimated 31 month FSD.

(2) The SLOC for the PC architecture was estimated to require 24% rewrite/redesign. This assessment was based to a large degree on the effort to incorporate the previously existing ISS program. Using similar relationships, PRICE S projected the cost of the software development for the "standalone" concept to be about $11.7M. If the software was completely rewritten, the cost was estimated to be $32.1M.

(3) For the networked architecture, the approximate 36% redesign/rewrite indicated a cost in the neighborhood of $16.3M. For a complete rewrite, the estimated cost was $23.7M.

(c) Case 2: Cost of Operation. The AOTS prototype demonstrated that advanced training and management technologies could automate an essentially manual system. Case 2 required many more assumptions to compensate for parameters that had not been determined and documented in a Statement of Operational Need (SON). Certain desired relationships such as the number of trainees per training terminal, the number of "contact" hours for the job-site environment, the number of tasks suitable for computer based training (CBT), the manner by which geographically separated units (GSU) would operate under AOTS, etc., were assumed and thereby increased the risk associated with the cost estimates. The operational costs were driven by the number of locations that a MAJCOM was spread across, the ratio of trainees per AOTS terminal, the amount of job-specific courseware developed and subsequently maintained for the AFSs within a MAJCOM, the relative ratio of complex-technical AFSs with non-complex, non-technical specialties within the command, and other performance factors attributing to the normal OJT process.

(1) The cost of identifying and developing the necessary courseware was estimated by poling subject matter experts, other research, and available commercial literature. To summarize, the courseware algorithm in the model allocates a range of 1-5% of the total tasks to CBT courseware depending on the AFS

type (ie., for highly complex, highly technical AFSs (type 1) the model uses 5% of the total training tasks in the courseware cost calculation). The dollar value of "development hours per CBT hour of instruction" was derived from comparing commercial literature with the report from a survey conducted by an AFHRL team that traveled to various Government sites engaged in computer aided instruction. The common finding was that an average development time per hour of finished contact time was difficult to defend because each lesson was dependent upon numerous variables that could not be averaged. However, 8 CBT modules were developed for the prototype AFSs at Bergstrom AFB and Ellington ANGB. Assessment of the hours spent by the AFHRL subject matter experts in preparing these CBT modules allowed us to derive the following range of courseware assumptions:

- Type 1 AFS: 200 development hours per 1 hour CBT module
- Type 2 AFS: 175 hrs per 1 hour CBT module
- Type 3 AFS: 150 hrs per 1 hour CBT module
- Type 4 AFS: 150 hrs per 1 hour CBT module

(2) The ALCCM assumes that an agency will be established to support the production software for all users. Hence, only one MAJCOM will need to budget for the software support effort. However, all using MAJCOMs were assigned the support costs of their AOTS specific hardware and the courseware. The following support costs were calculated by the life cycle version of General Electric's proprietary PRICE family of models, PRICE LC.

(aa) For the "mainframe" concept, PRICE LC estimated the annual software maintenance cost to be $1.63M.

(bb) For the "PC standalone" concept, PRICE LC estimated the annual software maintenance cost to be $1.39M.

(cc) For the "networked" concept, PRICE LC estimated the annual software maintenance cost to be $1.97M.

(3) Costs of operating an AOTS included the costs of data transfer. The latest DCA guidance to users of the DDN states that users will be charged for use of the DDN system in 1990. Hence, ALCCM uses the DCA provided cost algorithms to calculate DDN costs based on the expected number of "packets" of data. The model currently has a default value of 1 kilobyte per airman per month to calculate the number of data packets for the expected data transfer. This default was derived from assessing the present data flow for a "training status message" under the automated personnel data system (APDS).

3. Benefits. The AOTS prototype proved that the application of computer technology could provide significant benefits to the Air Force. Chief among these benefits were the automation of the system, the reduction in administrative costs, the standardization of training and force management and an increased force management capability. The on-going SLT&E is still inconclusive. However, indications are that as the users overcome a tendency to resist change, the system is perceived to be more time efficient, track OJT processes more accurately, and, interestingly, enhance the awareness of the importance of OJT in general.

(a) Some of the primary benefits achieved in automating the OJT system were determined by surveys. These include a more thorough understanding of the concept of "task analysis" (to assist in determining what has to be trained, when enough is enough, what constitutes "pass/fail" in job-site functions, etc.). Another item revealed in the surveys was that the more the system is used, the more value it has to the user. Although this appears to be "common sense" to those who are "computer literate", computers in general invoke fears in many people and these fears have prevented some from considering the usefulness of such devices.

(b) AOTS benefits are still being assessed in 5 basic areas: manhour savings (ie.,changes in hours spent by the existing labor force), manpower savings (ie., reductions in the required personnel), cost avoidances, value added, and changes in efficiencies.

4. Cost/Benefit Analysis. The results of the Cost/Benefit analyses are incomplete. These results will be available at the end of the program and published in October 1989. The paragraphs below describe the general nature of the on-going analyses:

(a) Manhour/Manpower savings. The surveys mentioned above are being assessed for reductions in manhours expended on the various tasks associated with OJT. Using standard Air Force labor rate factors (AFR 173-13), the manhours are converted to dollars, then annualized to postulate yearly savings. The costs of AOTS can then be amortized as a function of these annual savings over the expected life cycle.

(b) Cost avoidances. Various metrics are being considered. For example, certain printing costs will no longer be required by the elimination of forms that would be replaced by AOTS. TDY costs to interim technical schools would be reduced, and similarly, fewer requirements for ATC's Field Training Detachments (FTDs) will be required under AOTS.

(c) Value Added. Although it is difficult to measure the "value" of performing a mission "better", there are certain intrinsic characteristics that AOTS will enhance. For example, the following areas will enjoy an improved degree of standardization: OJT documentation, training requirements identification, training delivery techniques, and training assessment techniques. More training flexibility will be available under AOTS; increased unit readiness, or at least the documentation to substantiate the unit's "readiness" will be available to war planners. Preliminary analysis of the questionnaires indicates a positive response to AOTS, which could add to higher job satisfaction and higher retention rates. In short, AOTS adds value to the OJT process by offering more information to more decision makers than is currently available.

(d) Efficiencies. The analysis here is focusing on determining if AOTS indeed contributes to fewer paperwork requirements and fewer interruptions to the operational mission for job-site training. Additionally, this analysis is assessing increases in the Commanders' involvement in the OJT process, increases in the supervisors control over the training process of his/her people, and improvements in the use of the right people for the right job.

E. Supportability. Unlike the prototype, which was designed primarily as a proof-of-concept tool, the FSD design for AOTS must accommodate both hardware and software supportability goals. Maintainability and reliability goals for AOTS hardware are irrelevant because its design is based upon using existing COTS hardware. This is true whether AOTS is implemented as a standard system much as CAMS, PC III, and SPAS are being implemented, or if AOTS is implemented through an FSD program and becomes one of the many standard systems. Specifying maintainability and reliability criteria not readily available in the commercial world would significantly increase development costs without providing a corresponding reduction in life cycle support costs. Therefore, the primary hardware supportability issue becomes who will provide the hardware support and how will it be done?

1. Software Supportability. This tie to commercially developed systems is not the case for the software. The software is specially designed and developed for AOTS, and well defined software reliability, and maintainability goals will significantly reduce life cycle support costs while improving availability. Whether or not the Air Force chooses to implement AOTS as standard system or integrate its functionality into the existing standard

systems is again largely irrelevant. A function of any proof-of--concept program is to concentrate on demonstrating the capability and not place the same emphasis on the structured methods and "ilities" as you would during FSD. Unfortunately, unless reusability is emphasized at the initial stages of development, this approach usually yields a product that is not supportable and requires significant redesign to expand or enhance. While any software can be maintained, the amount of rewrite required to convert a prototype system to an FSD system must be carefully assessed against the cost of simply rewriting. The need for a careful assessment is the case with the AOTS software.

2. Software Supportability Analysis. A complete detailed software supportability and risk analysis of the prototype AOTS code was accomplished. This report basically concluded that several fundamental enhancements to the existing documentation and code were required before it could be considered transportable to other systems. It did not find fault with the overall design, but concluded that additional documentation would be required to effectively support the software. Such things as Detailed Design Documents, and Software Support Plans do not exist, and the existing Version Description Document and C-5 specifications were inaccurate or not in sufficient detail to support requirements traceability. The study did not conclude that the existing AOTS software was unmaintainable. Rather, it suggested major improvements to the software product characteristics prior to reusing the existing code in an FSD effort.

3. Hardware Supportability. Hardware support for software based systems executing on COTS equipment has traditionally been executed contractually by using commands who budget and fund for contractor maintenance. This is not always the case, but exceptions to this rule have generally been declared to be mission essential under AFR 26-1 and require Air Force organic (blue suit) support. For multi-command use systems such as AOTS (e.g. the World Wide Military Command and Control System (WWMCCS)) a single agency is usually designated as responsible for letting the hardware maintenance and support contract. Using commands then access the contract as required. This hardware logistics support posture is most likely for the production AOTS. Organic support is cost prohibitive and cannot provide for the multiple different hardware configurations envisioned in the Operational Concept Document. Any of a number of agencies could be designated as the contracting agency.

4. Support of Training Material. The Computer Aided Instructional (CAI) modules for delivering training as well as the test question and evaluation data bases, collectively called courseware, are not an integral portion of AOTS. In the same

manner, the MTLs, GPTRs, OPTRs, and OTR data bases are necessary to use AOTS but are not individually required portions of AOTS. AOTS provides the capability to develop and/or access these modules and data bases, but the actual end items themselves are not provided as part of the capability. AOTS software includes the software necessary to develop and maintain all this material; it is the responsibility of the Air Training Command, Functional Area manager, MAJCOM or some other entity to actually develop and maintain the items. The appropriate responsible organization will maintain ownership responsibility of this material and, after development or maintenance, route them through the AOTS system manager who will have configuration control responsibilities.

F. Technology Availability. The FSD AOTS will be built using existing technical capabilities available on the commercial market. As the system evolves, it is intended to continue working in the Air Force standard system environment.

1. Current Capabilities. At each level AOTS should use computers and peripheral devices that are standard Air Force equipment or Commercial Off-The-Shelf (COTS). AOTS will be designed to function primarily with what is available at the appropriate level. At the lowest work center level, it should function on personal computers such as the current Air Force standard Z-248 while at the base level it should function on the Air Force standard mini-computers and other standard work center computers such as the new Air Force standard AT&T 3B2. Peripheral equipment should, in most cases, be whatever is available at the level to support the host computer. In some cases, additional equipment such as expansion memory, disk drives (including floppy), optical character readers, modems, and network interface units will be required to meet minimum configuration requirements for the specific computer and level. In all cases, this additional equipment should be COTS and meet current DoD, Air Force and/or American National Standards Institute (ANSI) standards. System growth and upward/lateral compatibility should be the prime considerations.

2. Standard Computing Environment. AOTS should be designed to take advantage of advancing technology as the Air Force standard systems environment evolves. It should be designed and developed using established or emerging DoD or Air Force standards or in their absence, ANSI standards. This adherence to standard systems will allow the upgrading of systems to take advantage of new technology without becoming tied to specific systems. The full set of functions that an operational AOTS should provide must be viewed from a top level perspective. This perspective is provided in the Air Force Communication-computer Systems Architecture (AFCSA) series of documents. The AFCSA series provide technical

guidance, direction and proscribe standards for the employment of communication-computer systems (C-CS) in the Air Force. The primary purpose of the AFCSA is to provide a technical framework which allows Air Force-procured C-CS to operate in a "system of systems" environment. Some of the existing or emerging standards that must specifically be considered are as follows:

(a) One of the most pertinent of the AFCSA documents is Volume IV, Local Information Transfer Architecture (LITA). This document describes the existing base level computing environment, depicts a target architecture for future local information transfer systems, and outlines transition strategies to achieve that target.

(b) Requirements for long haul data communications should be accommodated by the Defense Digital Network (DDN). This is the DoD worldwide packet switching data communication network operated by the Defense Communication Agency (DCA) and imposing numerous standards on the user. Packet switches are essentially microprocessors designed for unattended operation using the Interface Message Processor X.25 standard protocol. Additional standards that apply to DDN users include the MIL-STD 1778, Transmission Control Protocol (TCP) and MIL-STD 1777, Internet Protocol (IP) for network to network transfer, routing, and precedence. Many TCP/IP limitations have motivated DDN to migrate to using the International Systems Organization (ISO) 7-layer Open System Interconnect (OSI) reference model.

(c) The Air Force Unified Local Area Network Architecture (ULANA) is the standard for a family of local area networks (LAN). The major functions of ULANA are to provide standard components for Air Force LAN implementations, to provide interfacing and interoperability among heterogenous hosts, work stations, and other devices, and to provide COTS components based on several DoD standards. The LAN distribution systems (e.g., fiber optics, coaxial cable, shielded broadband, twisted pair) are not presently part of the standard.

(d) The Government Open Systems Interconnection Protocol (GOSIP) has been established to allow information transfer between various local area networks such as ULANA and long haul services like DDN. This standard applies to the procurement of all mini or mainframes connected as intermediate or end systems as well as microprocessors or other personal computers that are part of an interconnected system such as AOTS.

(e) The follow-on standard Air Force mini computer should be a multi-processing machine using a UNIX type operating system. Since UNIX is a proprietary system, the IEEE proposed

Portable Operating Standard Interface for Computing Systems (POSIX) has been nominated for connections to a UNIX-like operating system. POSIX specifies the kernel interface to the computer but does not specify a standard method for accessing network services.

(f) The Ada computer language, MIL-STD 1815A, was used for the prototype and should be used in the FSD program. This language should allow greater portability between the various computers considered than any other language. It should also allow greater flexibility for modification and reduce the life cycle software maintenance cost.

(g) Adequate documentation is an essential portion of a system that is intended to grow and develop. Because of the reliance on standards in the code and interfaces, it is even more important. To insure adequate documentation, DoD-STD 2167A should be tailored to specifically identify and put on contract all necessary information.

SECTION III - TRANSITION STRATEGY

There are numerous approaches possible for transitioning AOTS from the prototype system to "full up" implementation in the Air Force, Air Force Reserves (AFRES), and Air National Guard (ANG). One approach might be to establish an FSD program through standard Air Force requirements and approval process (AFRs 57-1, 700-2) and the Planning, Programming, and Budgeting System (PPBS). Another method might be to transition the AOTS prototype technology to the managers of Air Force standard systems that already exist or are in development for implementation within their respective SPO. A third approach could be a hybrid of the first two -- implement selected AFSs through an FSD program and implement AOTS' (or portions of AOTS') technologies for other AFSs through modification of the appropriate standard system. Each methodology has distinct advantages and disadvantages. They will be discussed in the remainder of this plan. It is broken up into three parts. The first will deal with the overall approach, and schedule of the various strategies including implications of the various acquisition processes (AFR 57-1, 700-XX, 800-XX) and the PPBS. The second will deal with the Air Force management implications of the different strategies including some of the impacts on various Air Force organizations. The third part discusses acquisition strategy for an AOTS FSD program should the Air Force choose to implement AOTS that way. Possible acquisition strategies for implementing AOTS through standard systems are not discussed because they are the responsibility of the standard systems' program managers.

A. Transition Opportunities. The three possible architectures for the production AOTS are: 1) terminals controlled by a mainframe computer, 2) personal computer (PC) type work stations, or 3) networked PCs serviced by a centralized mini-computer file server. Regardless of the selected architecture, there are two approaches (along with a hybrid combination of them) for developing AOTS: 1) Full Scale Development as a standard system and 2) insertion of AOTS technology though standard systems. Additionally, regardless of the chosen development method, there are three ways to implement AOTS for wide scale use in the Air Force: 1) by MAJCOM, 2) by functional area communities such as Logistics or Security Police, or 3) Air Force-wide. Each of these alternatives are discussed below and recommended courses of action suggested. Transcending the architecture, method of development or implementation selected, the system requirements must be established and stabilized as early as possible. For example, a mobility requirement for the AOTS capability would obviate use of a "mainframe" architecture.

1. Architectures. As stated earlier, the three architectures investigated were terminals off a mainframe, a PC

based system, or a PC based networked system. In Appendix B, the requirements and effort to produce each of these architectures is described. The relative costs of these three architectures were discussed in Section II.D. As stated earlier (Section II.C.) there are significant advantages and disadvantages to each of the three possibilities. The main advantage of the mainframe architecture is that the prototype was built this way. A PC based system provides the greatest flexibility and mobility. A networked PC system blends most of the advantages of both and offers fewer disadvantages. When the total costs, advantages and disadvantages of each are considered, the best possible architecture for AOTS is a networked PC based system. This architecture appears to provide the greatest flexibility and growth potential at the least cost. When the selection for the follow-on system is made, the selection criteria should be decided as soon as possible and the three architectures reevaluated.

2. Development Approach. Either of the two approaches for developing an operational AOTS could use any one of these three architectures. If the follow-on program is based upon the Technology Insertion approach, selected functions could be implemented on the existing base level UNISYS 1100/60 with its already connected work stations. Other functions could be implemented on such standard systems as the Security Police Automated System (SPAS) where a mini-computer and PC based network service a base. The major risk associated with inserting AOTS technology in this manner is the potential for losing common functionality between systems because now the AOTS functions selected for implementation would be those that the System Administrator for the standard system deems appropriate. At this point, AOTS would no longer be a standard method of delivering and monitoring OJT but simply an extension of the other systems capabilities. This could also pose severe long term problems for the Air Force as standardization of the OJT system Air Force-wide is lost. On the other hand, if a Full Scale Development (FSD) approach is used, any one of the three architectures could be used, but the AOTS program manager would select the functions to be incorporated in the production system and Air Force-wide standardization would be maintained. This approach is discussed in detail in paragraph C "Acquisition Strategy" below.

3. Implementation Method. A third factor influencing AOTS transition to operational use in the Air Force is the implementation method selected. This selection is indifferent to the architecture selected or development approach. This is the portion of the transition that is concerned with taking a package, no matter how it is wrapped, and opening it up to service the Air Force. The three methods that could be used are: 1) by MAJCOM, 2) by AFS grouping or community, 3) or by directly institutionalizing

AOTS Air Force-wide. If AOTS is implemented by MAJCOMs, there is a distinct risk that the normal flow of personnel from assignment to assignment would produce a high rate of converting training records back and forth from Air Force Form 623s to automated records. This is very undesirable and poses the almost certain outcome of losing personnel training requirements in the conversion process. It would also significantly increase the training managers workload and be aggravated if OJT standardization is lost. These two problems together would probably completely negate any benefits derived from AOTS. Implementing AOTS Air Force-wide has the significant problem of sheer volume to contend with. However, Air Force wide implementation could be done incrementally by starting with a few bases in selected AFSs or groups of AFSs (communities) and gradually expanding to incorporate the same communities on other bases. As more and more bases come on line with these AFSs, other communities could be incorporated. This phased in community approach has the advantage of providing lessons learned, a significantly reduced risk of losing personnel "through the cracks", and would allow the Air Force to maintain a standardized OJT system.

4. Selection Decision. As the above discussion indicates, opportunities for transitioning AOTS from the prototype to a fully operational system are many. There is no one clear and easy path to achieve the benefits that the prototype proved can be obtained. The single most important point is that the decision on how to proceed must be made early in the transition process and then the requirements necessary to implement that decision must be determined and stabilized.

B. Management Implications. AOTS implementation will generate several top level management issues that need to be resolved prior to the transition to an operational system. These issues primarily fall in the realm of organizational relationships and how the program will be funded. The first of these, Organizational Roles, must be at least partially resolved during the decision process outlined in paragraph A above. The second, Funding, is a concern that must be addressed regardless of the transition implementation decision but should be at least partially resolved during the decision making process.

1. Organizational Roles. The management implications that must be clearly defined and documented fall into six categories which are: 1) system acquisition responsibility, 2) software and hardware support responsibility, 3) OJT task list management, 4) ancillary and other training list management, 5) training records control and management, and 6) instructional courseware development and management.

(a) The system acquisition responsibility has to be defined to designate who is going to be in charge, make the decisions, obtain the funding, develop the system and define operational responsibilities. Because of their extensive experience with the AOTS prototype and their repository of knowledge in human resources and training, the acquisition agency should be AFSC/HSD. A FSD system acquisition program should use a careful blend of AFR 700 and 800 series regulations with the emphasis being on the 700, automated data processing, side. Because of the possible significant impact on the Air Force as a whole, the implementing directive and initial decisions for the program should originate out of the Air Staff.

(b) Support responsibility for the AOTS has to be defined in terms of both the software and the hardware. Both of these should be at least partially resolved during the implementation decision process. If a FSD program is instituted, then a Software Support Center (SSC) needs to be designated responsible for follow-on support and made a part of the development process. This should be an Air Force Communications Command (AFCC) organizational responsibility if AOTS is to be implemented Air Force wide. This SSC would be responsible for supporting the algorithmic and instructional support system software as well as the interface software necessary to allow AOTS to tie into other Air Force standard systems or instructional courseware developers. The SSC would also have configuration management and control responsibilities. Hardware support responsibility would be determined by the method of implementation. If AOTS becomes a standard system, then an Air Force Logistics Command (AFLC) organization should provide a "standardized" multi-user hardware maintenance and support contract. On the other hand, if AOTS becomes part of another standard system, or is hosted on a standard system, then that system's hardware maintenance and support concept would prevail.

(c) Air Training Command (ATC) should continue responsibility for maintaining the Master Task List (MTL) of OJT requirements in conjunction with the AFS Functional Area managers and the Occupational Measurement Center. Modification and control of these lists would remain essentially the same as the present system with the additional responsibility of maintaining configuration control of the AOTS formatted listings. This AOTS formatting process will result in a greatly expanded task list that under the present system is condensed on the Air Force Form 623.

(d) MAJCOMs and other headquarters organizations down to the unit and individual supervisor will have designated ancillary and OTR list management responsibilities. Each echelon in the chain has its role to be filled in defining the total training requirements that the fully operational AOTS will track.

Essentially training requirements for AFS job tasks and OTRs for individual duty positions within the unit will be defined. The supervisor will be responsible for fitting the individual to the duty position and using the OPTR to define the training required.

(e) Unit commanders, training monitors and supervisors will be responsible for maintaining the individual ATR. This ATR will consist of all training requirements regardless of what level was specified and be a record of the individual's actual training.

(f) Instructional courseware should be developed for the AOTS by ATC, Functional Area managers and MAJCOMs. The AOTS program manager (system administrator) will be responsible for maintaining the interfaces and configuration of the training development and delivery portion of AOTS that will allow users to develop courseware. The organization requiring the courseware will be responsible for its development by any means they desire that meets interface and integration requirements established by the AOTS program manager. Responsibility for certification and control of courseware is an open issue. Certification should be delegated to the functional area managers at the lowest level possible. Certified courseware could be controlled and distributed by ATC in conjunction with Functional Area managers.

2. Funding. The decisions about funding will be a part of the decision on how to implement AOTS. If AOTS is implemented using the technology insertion approach, then the primary source of funding for software development and any necessary hardware acquisition should be shared by the developing agencies and the using agencies. Operations and Maintenance (O&M) funds commonly designated as 3400 funds can be used to augment the developing agencies' budget and the using agencies would provide for installation using appropriate command allocated funds (3400 or 3080). If a FSD program is used, then the primary source of funds will be through a Program Element (PE) established under the Programming, Planning and Budgeting System (PPBS). This would be the preferred method because AOTS can be inserted as a line item in the Program Objectives Memorandum (POM), and compete on its own merit with other Air Force requirements. It also is probably the only methodology that permits orderly planned implementation of AOTS on an Air Force wide basis. Funds for the development activity, commonly called 3600 funds, would allow for the establishment of a SPO and the full scale development of an operational system. Implementation would be funded with both 3080 (central procurement) and 3400 funds O&M. The SPO could also be charged with planning and accomplishing the actual implementation. A variation on using the PPBS approach would be to use a PE for the FSD of an operational system then rely on MAJCOMs to "buy AOTS" by funding for

implementation using their command allocated 3400 and 3080 funds. These installation costs could be estimated by the SPO and installation, MAJCOM and Functional Area managers required to budget for them. Funding for courseware development for use by AOTS would be the responsibility of the requiring agency.

C. Acquisition Strategy. This paragraph addresses some of the key issues that must be resolved prior to contracting for the development of a follow-on fully operational AOTS. It assumes that the Air Force has decided to develop an AOTS capability through a Full Scale Development effort. It further assumes that a generalized System Operational Requirements Document (SORD) has been developed and approved, and that the FSD activity is required to develop a system easily expandable to the entire Air Force. It further assumes that while generalized requirements have been established through the SORD, detailed requirements and interfaces with other standard systems have not been done. It also assumes that the development, initial installation(s) and testing (both DT&E and IOT&E) of the operational AOTS must be completed prior to Air Force acceptance from the development contractor.

While the discussion assumes that an implementing directive (Program Management Directive or Information Services Directive) has been issued and a cadre responsible for the FSD program has been formed, it does not assume that the implementing command is AFSC. This can be done because most of the activities the cadre must complete prior to finalizing a Statement of Work (SOW) are common to the acquisition process regardless of the implementing guidance (AFR 700 series regulations vs AFR 800 series regulations) or the implementing command. Most of the activities that the cadre must complete prior to contracting are discussed below. Since there are numerous decisions that need to be made prior to actually developing a full acquisition strategy, the following is not designed to provide the acquisition strategy; rather, is designed to step through the major steps that need to be accomplished.

1. Program Management Plan. The very first objective the cadre must accomplish is to establish a detailed Program Management Plan (PMP). While this will be a difficult and time consuming task, it must be completed early enough to set the stage and tone of the acquisition. Development of the PMP will require the establishment of working groups with the initial user and other affected parties such as ATC, USAF/DP, AFLC, and other MAJCOMs that are or will be affected by an AOTS. Program managers for other standard systems should also be included in the development of the PMP because of the impact that AOTS may have on their systems. The plan should be coordinated with the affected agencies and as a minimum define the participants' roles and responsibilities towards the FSD program as well as their relationships with each other.

This plan should define the objectives of the FSD program, its funding, and schedule. It should also define and establish the various inter-agency working groups (e.g. Business Strategy Panel, Interface Coordination Working Groups, Computer Resources Working Group, Test Planning Working Groups, etc.) that will be required to support the FSD program.

2. Requirements Definition. Detailed definition of requirements is almost equally important to developing and coordinating the PMP. The issuing directive (Information Systems Directive (ISD) or Program Management Document (PMD)) and SORD will have already established generalized requirements, but not in sufficient detail to allow execution. Numerous decisions must be made. How much of the AOTS data base, if any, can be off-loaded to another standard system? How will communications be maintained to ensure that common data between another standard system and AOTS do not diverge over time? For example, AOTS records contain certain personnel information that is also contained in PC III.

(a) Several trade-off studies also need to be done prior to establishing the "A Specification." Should the prototype code be rewritten? Should Ada be the design language or should a COTS Data Base Management System (DBMS) be used as the engine for the data base management activities augmented by Ada code whenever the COTS DBMS can not efficiently provide the required functions? What are the design criteria? Should the product of the FSD be a software package that is usable on several different kinds of PCs or should AOTS be optimized against a single machine architecture and/or a single operating system?

(b) Security implications must also be addressed. This applies not only to such things as classified information, but also to personal privacy. The prototype AOTS contains information that is covered under the Privacy Act such as social security number (SSAN), age, etc. How will access to that data be safeguarded if it is also required in the operational AOTS?

3. Implementation Strategy. Strategy for implementing AOTS in work centers at Air Force bases must also be established prior to contracting the FSD effort. How many AFSs will be serviced by the FSD product? What courseware must be developed prior to implementation? What is the implementation schedule for the Air Force? How many bases and how many AFSs will be included under the FSD activity? How will affected work centers and training records be converted to the automated system? How will training records be changed back to the manual system for those airmen transferring from an AOTS serviced work center to a non-AOTS serviced work center? Who will be responsible for implementing additional bases/AFSs after the development contractor has

completed the initial installations? What data is required to permit a second party to install AOTS in a new work center or on a new base after the initial installations by the development contractor?

4. Business Strategy. An acquisition plan must be developed and prior to that the business strategy established. This is normally accomplished in coordination with affected agencies, the contracting authority, and the FSD program manager and approved by a Business Strategy Panel. Competition Plans must be written and potential contractors must be identified. This can most easily be done through market surveys using Request for Information in the Commerce Business Daily and draft Requests for Proposal. Contracting strategy must be defined. This will include such determinations as contract type (cost plus fixed fee, cost plus incentive fee, fixed price, etc.) and whether or not multi-year contracting is appropriate. Source selection procedures in accordance with AFR 70-30 must be developed and approved by the source selection authority.

D. Acquisition. It is only after the decisions outlined in paragraphs III A, B, and C above have been made that the full acquisition strategy for the follow-on system can be developed. Each of the stages in the decision making process will affect the overall transition strategy and without these decisions, no single strategy can be used. On the other hand, as the decisions are made, the required transition and resulting acquisition strategy will be developed.

APPENDICES

Appendix A. Prototype System Description. This Appendix describes the AOTS prototype in detail.

Appendix B. Prototype System Evolution. This Appendix describes the changes and evolution necessary to make the AOTS a fully operational system.

Appendix C. Glossary. Terminology as well as acronyms.

Appendix D. Deliverables. A comprehensive list of deliverables from the prototype program.

Appendix E. References. A listing of referenced regulations and documents.

APPENDIX A AOTS PROTOTYPE DESCRIPTION

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SECTION I - FUNCTIONAL DESCRIPTION

For description purposes the AOTS is grouped into three functional components: 1) management, 2) training development and delivery, and 3) evaluation. The management functions are those that identify AFS tasks and other training requirements, and manage and certify their accomplishment. A Master Task List (MTL) is defined for each Air Force Specialty (AFS) which contains all AFS-related tasks that a holder of that specialty might be required to perform. The training development and delivery functions are those necessary to develop and deliver Computer Aided Instruction (CAI). They consist primarily of the adapted Instructional Support System (ISS) software for development and Interactive Video Disk (IVD) interfaces for delivery. The evaluation functions are those necessary to evaluate individual airman knowledge and proficiency.

A. Management Functions. The AOTS management function provides for the development and control of Master Task Lists (MTL), Generic Position Task Requirements (GPTR), Operational Position Task Requirements (OPTR), Other Training Requirements (OTR) and their refinement down to an actual Airman Training Records (ATR) and the Individual Training Requirements (ITR) list. The function also supports the planning of training materials, the development of training plans for individual airmen, and the creation of training schedules.

1. Training Requirements Management. Training Requirements Management function provides computer-based support for procedures to identify task knowledge and proficiency requirements and associated training requirements. The function also provides specifications for all performance and training requirements associated with an Air Force Specialty Code (AFSC).

(a) Master Task List. The management function controls the development and maintenance of a Master Task List (MTL) for each AFS of all job-related tasks a holder of an AFSC may be required to perform. This list includes all the general and specialized tasks dealing with any special identifiers and different equipment the AFS is associated with. The information for an MTL is developed and defined by the respective Functional Area Manager and the Air Force Occupational Management Center for each AFS and provided to AOTS. Subject Matter Experts (SME) would use this information to develop AOTS MTLs. An example would be all the tasks required to be performed by jet engine mechanics regardless of whether they are on a fighter, a helicopter or a piece of Aerospace Ground Equipment (AGE).

(b) Generic Position Task Requirements. Generic Position Task Requirements (GPTRs) are subsets of MTLs that deal with specific types of duties. The GPTR would narrow the jet engine mechanic's training down to a specific piece of equipment such as an F-15. Master Task List data are used to develop generic lists of training requirements for common position descriptions. Generic position descriptions contain position identification, task requirements, and training sequence information. Each GPTR constitutes a representative set of tasks which a duty position may be required to perform and contains a training sequence. The GPTR is a template for the Operational Position Task Requirement.

(c) Ancillary, Additional Duty, Other Training Requirements (OTR). Ancillary, additional duty and all other training requirements are identified. They are then grouped based on common relationships such as standard training or crew chief requirements. The OTR is the source of identifying the training required over and above the AFS specific training necessary to function in a specific position such as a flight line mechanic being assigned as a crew chief.

(d) Operational Position Task Requirements. An Operational Position Task Requirement (OPTR) lists the specific set of tasks which the person filling an operational duty position may be required to perform. Some operational position tasks and other requirements have been defined for manpower positions. The OPTR is developed by the supervisor from the GPTR and the OTR. The AOTS provides the supervisor with an automated environment to develop OPTRs that are focused in on all the tasks required of a specific duty position. The OPTR contains position identification, task requirements, task training sequences, and contingency, ancillary, and additional duty training requirements information.

(e) Airman Training Record (ATR). The supervisor then uses AOTS to copy the OPTR for a specific duty position to develop/modify the ATR for the airman holding that position. The four major categories of data maintained on the ATR are: personnel; training history to include formal, ancillary, task, and contingency training, and PME, additional duty, and career development courses; position qualification and task certification history; and task trainer qualifications.

(f) AFS Performance and Proficiency Function. The task performance and proficiency function establishes the specifications, source document references, and inter-task relationship data contained on the MTL.

2. Airman Training Management. Airman training management identifies the duty position to which an airman has been assigned, diagnoses the training required, manages airman progress

toward position qualification, and maintains training records reflecting airman training status.

(a) Airman Training Record. An Airman Training Record (ATR) provides a permanent record of all training received by an airman throughout the airman's career. The supervisor maintains the ATR by logging training and obtaining updated data from the personnel system.

(b) Training Needs Diagnosis (Qualification Assessment). AOTS identifies an airman's training requirements by comparing data provided by the position training requirements; contingency, ancillary, and additional duty; and career development course management with the airman's training history in his/her ATR. AOTS provides qualification assessment and position allocation. AOTS also supports adding requirements to an ITR. The AOTS allows altering, canceling, and recording an airman's attendance at a scheduled event.

(c) Individual Training Requirements. An airman's Individual Training Requirement (ITR) is the result of an automatic training needs diagnosis performed by AOTS. An ITR is derived by comparing an OPTR defining a position against the ATR of the airman assigned to that position. The AOTS develops the ITR on demand. The ITR contains such information as a record of all training required and in progress. The ITR record includes airman and position identification data, identifies individual training requirements, indicates airman ITR status, and provides airman training plans. It identifies the CDC and OJT requirements.

(d) Career Development Course Management. The AOTS provides the functions to manage Air Force Career Development Course (CDC) requirements, enrollments, and status. It obtains the requirements from the ATR during the training needs diagnosis.

(e) Trainee Progress Management. The AOTS manages trainee progress with measures such as trainee position qualification progress, trainee task certification status, and trainer performance record and scheduling.

(f) AFS Performance Evaluation and Training Requirements. The performance evaluation and training requirements function identifies and prioritizes requirements and assigns tasks to On-the-Job Training (OJT).

(g) Schedule Generation. The AOTS includes a scheduling management function that may provide computer-based support for scheduling training for one or more airmen, for scheduling evaluations, and for modifying training and evaluation schedules as required.

B. Training Development and Delivery Functions. The training development and delivery functions provide for the development and delivery of Computer Aided Instruction (CAI). This function is primarily provided by the Government furnished ISS software which was adapted and fully integrated into AOTS. IVD lesson development is also supported by AOTS as well as the deliver of IVD lessons that are stored at an AOTS work station.

1. Training Development. The training development function of AOTS provides the training developer with mechanisms or tools for producing and maintaining computer based instructional modules. The mechanisms provided will support training development in accordance with AFM 50-2, Instructional Systems Development.

(a) Objectives and Test Item Development. Task relevant data is analyzed to identify and validate the skills and knowledge relevant to task performance. Behavioral objectives have been developed for tasks identified for performance evaluation and/or job site training. Knowledge test items that have been developed are maintained in a test item bank.

(b) Instructional and Evaluation Materials Development. AOTS supports the integrated ISS computer aided courseware authoring system in the development of instructional and evaluation materials.

2. Training Delivery. The training delivery function of AOTS stores, distributes, and controls access to the instructional materials stored on-line by AOTS. This component also allows trainees to receive on-line training using the integrated ISS training delivery system.

(a) Access Control. AOTS provides the mechanism to control access to training and evaluation materials and privacy data.

(b) Assignment Generation. The AOTS provides capabilities to generate event assignments and schedules.

(c) Materials Storage and Distribution. The AOTS stores and distributes all on-line AOTS-compatible training and evaluation materials.

C. Evaluation Functions. The evaluation function supports assessment of trainee task proficiency and knowledge.

1. Trainee Evaluation. AOTS provides three functions to support performance evaluations. First, it allows trainees to take on-line knowledge evaluations, then score the evaluation, record the results, and provide the trainee with feedback on the results. Second, it prints evaluation forms for off-line evaluation and supports status tracking of the evaluations that have been printed. Finally, it supports scanning equipment that reads answer forms for off-line evaluations, scores knowledge test results, and records all results.

(a) Task Performance Evaluation. The AOTS supports performance evaluations upon the successful completion of performance task training.

(b) Knowledge Training Evaluation. The AOTS administers and processes knowledge training evaluations that were completed either on- or off-line.

(c) Performance and Training Evaluation Data Collection. The AOTS captures evaluation data such as completion of training, date, time, and location of training, individual item responses and scores, and trainer and evaluator identity.

2. Training Quality Control. This function supports the systematic task performance evaluation procedures that assess the effectiveness of training conducted under AOTS. The two primary quality control functions are training quality control evaluation selection and data collection. The selection function includes a sampling algorithm designed to select tasks, evaluation candidates, and qualified external evaluators. The data collection function records the results of the quality control evaluations for review and possible action by work center supervisors and unit commanders.

3. System Evaluation. The AOTS system gathers system data and generates standard and ad hoc reports.

SECTION II - SYSTEM DESCRIPTION

The AOTS prototype is a centralized software system operating under the VAX/VMS operating system. AOTS is predominantly written in the Ada programming language, but a collection of FORTRAN, MACRO, and SAS routines are also included. The AOTS host computer is connected to the operating locations by high speed fiber optic links, and users work interactively using Zenith Z-248 PCs as dumb terminals emulating VT-100s.

A. Hardware Architecture. The overall hardware configuration of the prototype AOTS consists of user work stations connected to a VAX 8650 computer system via high speed communications systems. Figure A-1 depicts the top level hardware architecture for AOTS. Also connected to the VAX are laser printers and other micro-computers/terminals which are used for development of AOTS software and courseware.

There are 34 AOTS user work stations at Bergstrom AFB and Ellington ANGB. A minimal work station consists of a Zenith Z-248 personal computer. A fully equipped work station includes the following equipment:

- Zenith Z-248 personal computer
- Alps P2000G dot matrix printer
- Scantron optical mark reader (OMR)
- Sony LPD-2000 interactive video disk (IVD) player

The Z-248 and the Alps P2000G printer are connected to the VAX system via the communication systems. There is no direct connection between the Z-248 and the Alps P2000G printer for local printing. When an OMR is present at a work station, the OMR is connected to the VAX system via the communication systems and the Z-248 communicates with the VAX through the OMR. The IVD is hardwired directly to and controlled by the Z-248. A typical prototype AOTS work station is illustrated in Figure A-2.

There are an additional 48 Z-248s that support the AOTS in the areas of prototype software development and Instructional Systems Development (ISD). The configuration of each development station varies.

A complete list of equipment that makes up the hardware system of the prototype AOTS, and a description of the major hardware components are provided in the following paragraphs. A description of the AOTS operating locations, facilities, and communications systems is also provided. For additional information refer to the Critical Item Development Specification for the

Hardware Component of the AOTS, Douglas Aircraft Company (DAC) Specification number 70S647401.

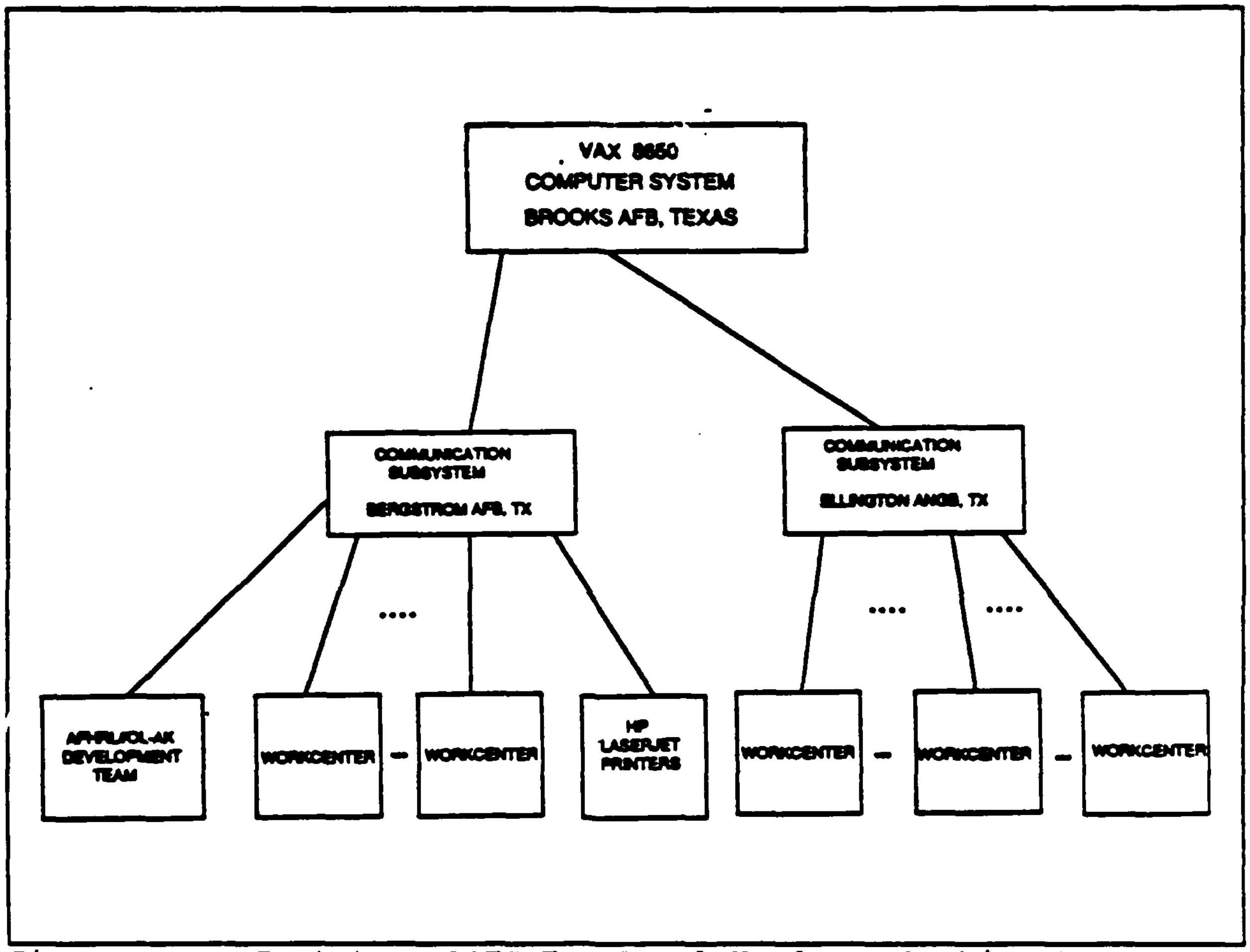

Figure A-1. Prototype AOTS Top Level Hardware Architecture

1. Equipment. Table A-1 and Table A-2 list the components that comprise the prototype AOTS development centers and work centers located at Bergstrom AFB and Ellington ANGB respectively. The AOTS software and databases are hosted on a VAX 8650 computer system at the AFHRL Computing Center, Brooks AFB TX.

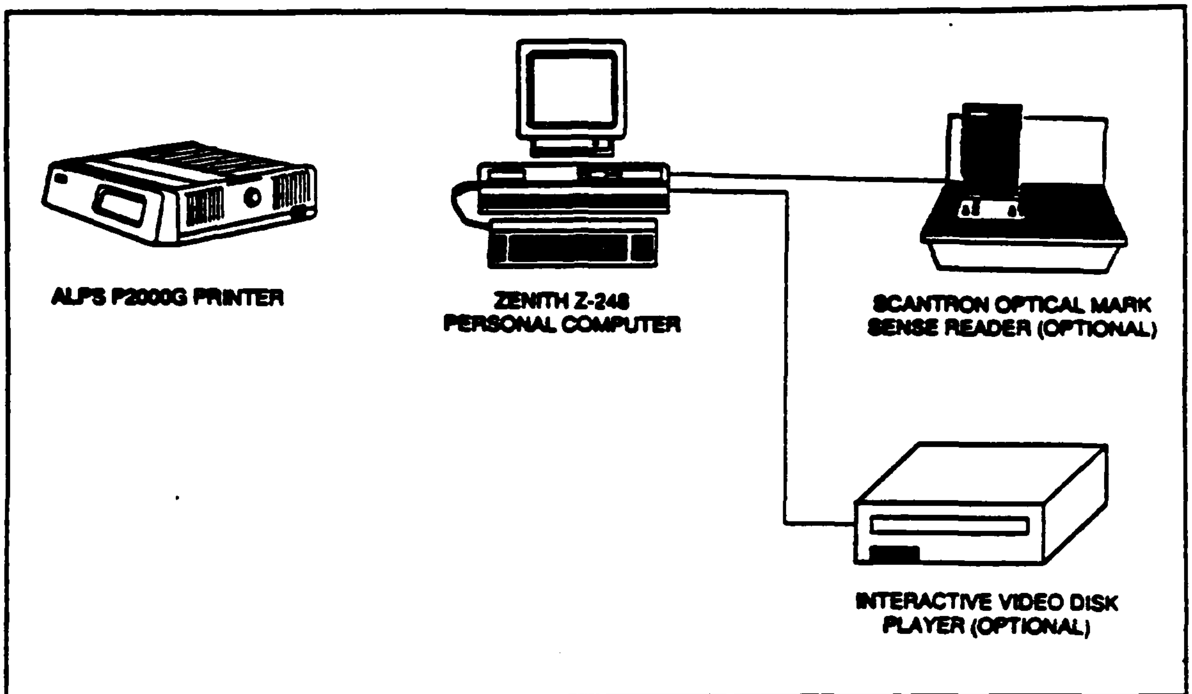

Figure A-2. Typical Prototype AOTS Workstation

Table A-1. Bergstrom AFB Prototype AOTS Equipment List

Equipment	Quantity
Zenith Z-248 Personal Computer	74
Alps P2000G Printer	21
Hewlett-Packard LaserJet Printer	5
Scantron 5200 Optical Mark Reader	14
Sony LPD-2000 Interactive Video Disk Player	12
Infotron 992NP Network Processor	3
Universal Data Systems Model 56 CSU/DSU	6

Table A-2. Ellington ANGB Prototype AOTS Equipment List

Equipment	Quantity
Zenith Z-248 Personal Computer	9
Alps P2000G Printer	9
Scantron 5200 Optical Mark Reader	5
Infotron 992NP Network Processor	1
Universal Data Systems Model 56 CSU/DSU	1

The Digital Equipment Corporation (DEC) VAX 8650 is a high-end, multi-user mainframe computer system. The central processor uses a 32-bit architecture with 4 gigabytes of virtual addressing space. The system has 16 megabytes of 256K ECC MOS memory. AOTS dedicated mass storage is provided by DEC RA81 fixed disk drives. Each fixed disk has 456 megabytes of memory. Backup storage is provided by TU81 tape drives. The computer system also houses a 56 kilobaud modem for high speed communications to remote terminals and printers. Additional information on the VAX 8650 computer system can be found in DEC VAX Systems and Options Catalogs and other applicable DEC documentation.

AOTS users interface with the VAX 8650 system using Zenith Z-248 series personal computers as terminals. The Z- 248 is based on the Intel 80286 16-bit processor operating at 8 MHz. A typical Z-248 computer in use by AOTS is configured with a Zenith ZVM-1380 RGB color monitor, an Intel 80287 math co- processor, EGA graphics, 512 kilobytes of main memory, a 20 megabyte Winchester fixed disk drive, and a single 5.25 inch floppy disk drive with a capacity of 364 kilobytes. The computers have two serial RS232-C interfaces and two parallel interfaces. Selected Z-248's have been configured with additional cards for interfacing with and controlling a Scantron OMR and/or a Sony IVD.

All AOTS work stations have access to an Alps P2000G dot matrix printer. The printer is used to obtain hard copy printouts including reports and notices generated by the AOTS. The printers are treated as terminal ports on the VAX.

Some AOTS work stations include a Scantron 5200 Optical Mark Reader (OMR). The OMR is used by evaluators to score off-line knowledge tests and performance evaluations. The OMR can scan forms of sizes between 3 x 5 inches and 8.5 x 14 inches and can read forms marked with a #2 pencil. The scan rate is 13 inches per second. The Scantron OMR has two RS-232C interfaces, each with rates of 75 to 19,200 bits per second. One port is available for connection to the Z-248 and the second port is available for optional connection to a CRT terminal. When an OMR is present at a work station, the OMR is connected to the VAX system via the communication systems and the Z-248 communicates with the VAX through the OMR.

Some AOTS work stations also include a Sony LDP-2000 IVD. The IVD is used to generate and playback audio/video oriented courseware. The IVD is connected to the Z-248 via an RS-232 interface connector. The IVD has a built in microprocessor that controls almost all functions of the player. Additional control of functions is provided by the personal computer. The IVD capability was added to AOTS late in the prototype development process. As a result, only a small number of work stations include an IVD player for demonstration purposes. The work centers with IVD's require color, touchscreen monitors. The monitors used with the prototype AOTS are 13 inch Electrohome ECM-1311U high resolution color monitors. These monitors were modified at the factory to accept custom cables and to add the touchscreen capability. The custom-made cable required to connect the Z-248 PC to the monitor were designed and manufactured by the HRL. The IVD is controlled by a Matrox controller board residing in the Z-248. These Matrox controller boards were procured under a separate Army contract.

Additional hardcopy printing capability is provided by Hewlett-Packard LaserJet 500+ series laser printers. Some HP LaserJet printers are connected to the VAX system. Other LaserJet printers are connected to development station Z-248 computers for local printing. The LaserJet prints using high-resolution dot matrix pattern (300 x 300 dots per inch). The printer has 512 kilobytes of RAM and seven resident fonts. Dual paper cassettes and automatic paper feed allow continuous, uninterrupted printing. Additional features include downloadable font capability, macro capability, and advanced graphics abilities such as shading patterns and various values of gray shading.

2. Facilities. The prototype AOTS hardware described in the preceding paragraphs resides in facilities at Brooks AFB, Bergstrom AFB, and Ellington ANGB TX. The VAX 8650 mainframe computer system is housed in the AFHRL Computer Center at Brooks AFB. Additional AOTS equipment installed at the AFHRL Computer Center includes an expanded Infotron 992NP Network Processor and Universal Data Systems Model 56 Customer Service Units (essentially a 56 kilobaud modem). The remaining AOTS equipment consists of development work stations and AOTS user work stations at Bergstrom AFB and user work stations at Ellington ANGB.

There are 26 AOTS user work stations located in 15 work centers at Bergstrom AFB. Seven of these work stations include IVD players. The user work stations provide service to the following Air Force specialty areas: Jet Engine Maintenance, Aircraft Maintenance, Personnel, and Security Police. Table A-3 provides a list of building numbers where AOTS equipment is installed and the quantity and type of equipment installed.

Table A-3. Bergstrom AFB User Workcenter Facilities

	Z-248	ALPS2000	Scantron
Jet Engine Maintenance			
Building 1612	2	1	1
Building 4529	1	0	0
Building 4589(RES)	2	1	1
A/C Maintenance			
Building 1609	1	1	1
Building 4529	2	1	1
Building 4515(RES)	2	1	1
Personnel			
Building 2202	3	1	1
Building 208	1	0	0
Building 4555(RES)	1	1	1
Security Police			
Building 207	1	1	0
Building 208	2	1	1
Building 4204(RES)	2	1	1
Building 1805	1	1	0
Base OJT Monitor			
Building 2202	1	1	0
Chief of Maintenance Training Staff			
Building 1501	3	1	1
Building 4592(RES)	1	1	1
TOTAL	26	14	11

In addition to the user work centers listed above, there are three AOTS development facilities located at Bergstrom AFB. These facilities support the software engineers, instructional software developers, analysts, and the Instructional Systems Team. Table A-4 lists the equipment located at each of these three development facilities. Additional information regarding AOTS facilities at Bergstrom AFB can be found in the AOTS Site Preparation Requirements and Equipment Installation Plan, DAC, dated 26 January 1988.

At Ellington ANGB, 9 AOTS user work stations are housed in four buildings and provide service to the following Air Force specialty areas: Security Police, Personnel, Jet Engine Maintenance, and Aircraft Maintenance. Table A-5 lists the building numbers where AOTS equipment is installed and the quantity

Table A-4. Bergstrom AFB AOTS Development Facilities

	Z-248	ALPS2000	HPLaser	Scantron	IVD
Building 428	17	1	3	2	0
Building 1808	30	5	2	0	5
Building T-1	1	1	0	0	1
TOTAL	48	7	5	2	6

and types of equipment installed. For additional information regarding AOTS equipment installed at Ellington ANGB refer to the Facilities Requirements Plan for the Transition of the AOTS to Ellington ANGB, dated 16 October 1987.

Table A-5. Ellington ANGB AOTS User Workcenter Facilities

	Z-248	ALPS2000	Scantron
Building 1193	2	2	1
Building 1057	2	2	1
Building 1293	1	1	1
Building 1382	4	4	2
TOTAL	9	9	5

3. Communications. The three AOTS operating locations (Brooks AFB, Bergstrom AFB, and Ellington ANGB) are connected by high speed (56 kilobaud) fiber optic communications lines and modems. The modems interface with the VAX and other AOTS equipment through Infotron 992NP network processors/multiplexers. As illustrated in Figure A-3 there are two 56K lines connecting Brooks AFB and Bergstrom AFB and a single 56K line connects Brooks AFB and Ellington ANGB.

Communications start at the VAX 8650 computer system on the DMZ32 boards. Each DMZ32 has 24 channels with a DB25 male connector on each. The current configuration at Brooks AFB is a 50 pair cable connected to the DMZ32's and running to the patch panel. From the patch panel, a cable is run to the 992NP. The 992NP is connected to the UDS CSU/DSU 56K modem with standard telephone wire (four wire). The 56K modem is connected to the 56K communications line.

The intrabase communication network at Bergstrom AFB consists of two 56K lines, three 992NP multiplexers (one with an expansion nest), and six UDS CSU/DSU 56K modems. One 56K line from Brooks is connected to Building 428 and the other line is connected to Building 1101. Building 1101 acts as the central communications node at Bergstrom AFB as illustrated in Figure A- 4. The 56K lines from Brooks are connected to the 56K modems which are in turn connected to 992NP multiplexers. The 992NP communication ports are connected to plug-on line drivers to boost the signal. Four wires are then run to the communication patch panel in building 1101 and then connected to the appropriate buildings via base communications. Building 1808 is connected to both Building 1101 and Building 428 via 56K communication lines.

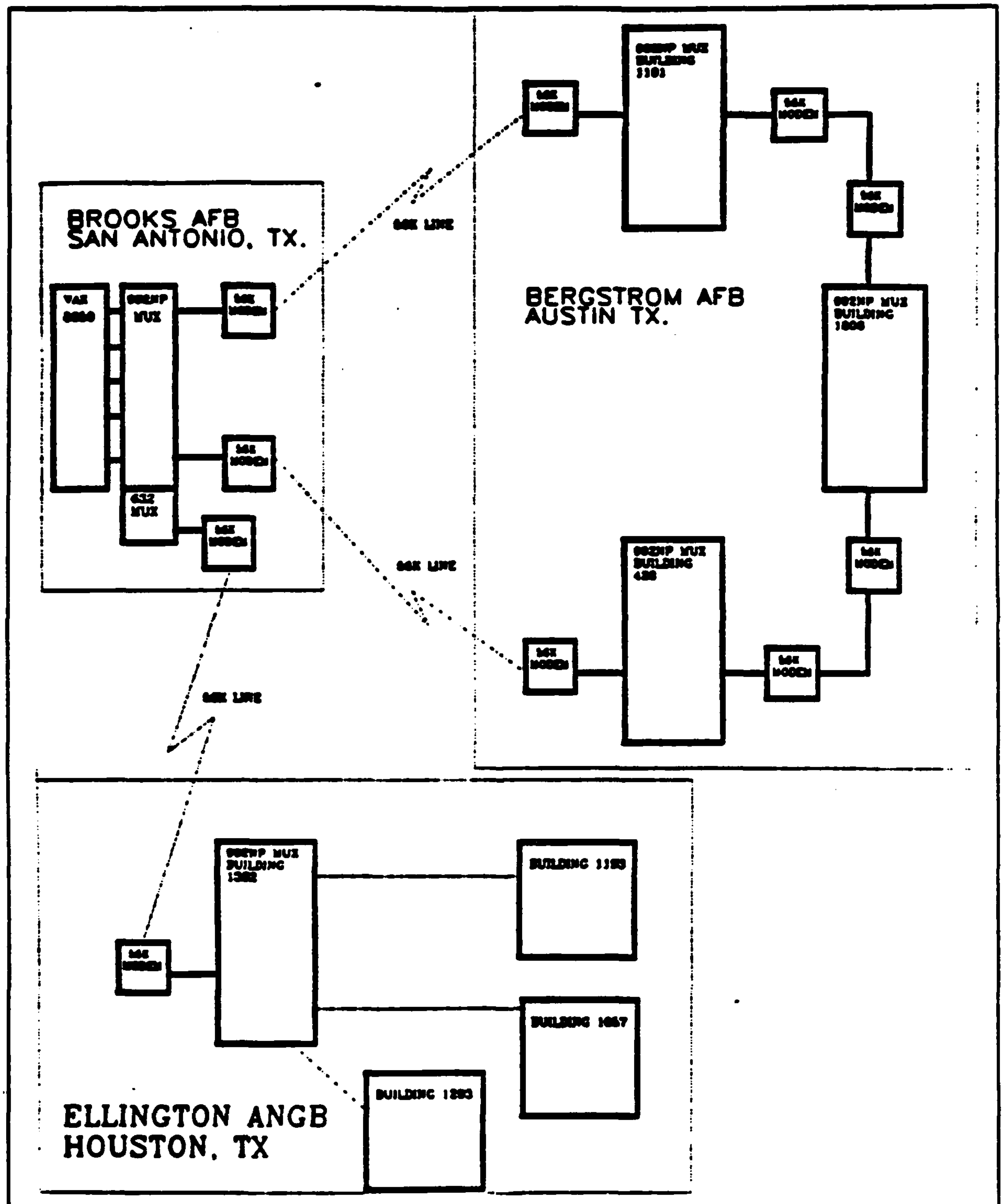

Figure A-3. AOTS Top Level Communications Architecture.

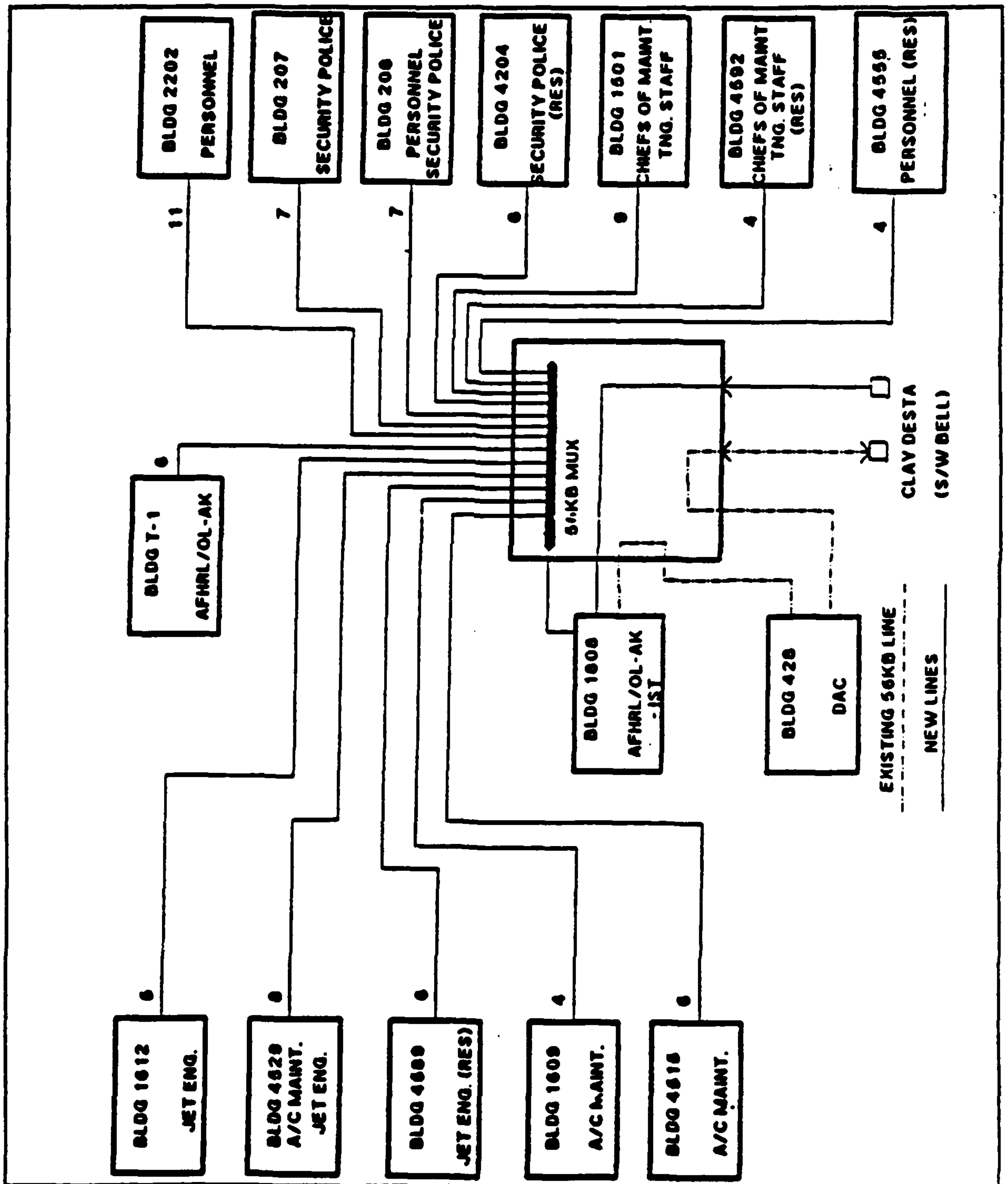

Figure A-4. Bergstrom AFB Communications Network

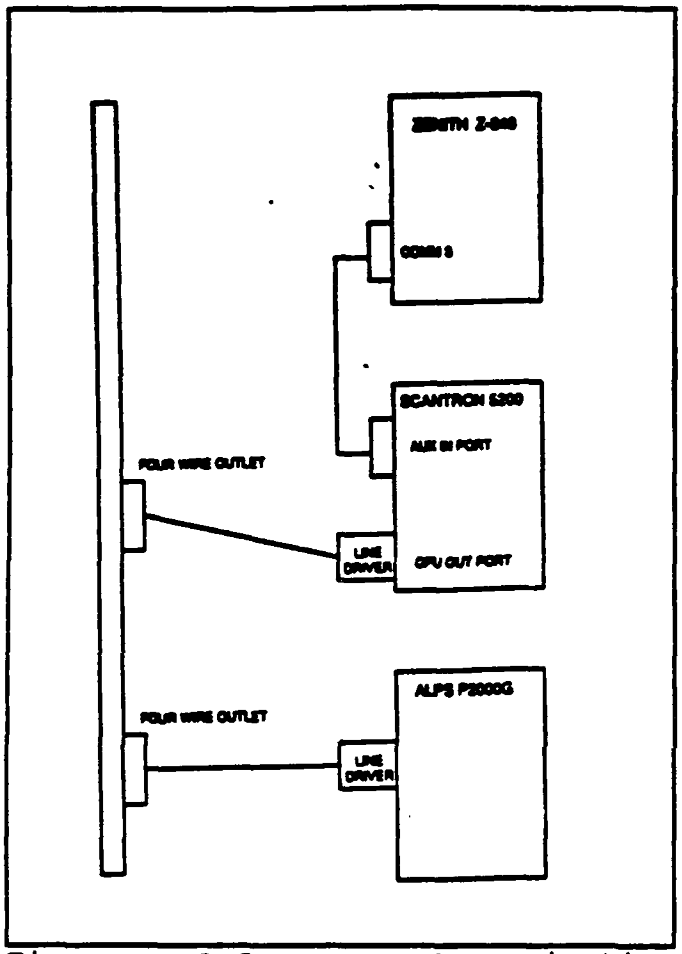

Figure A-5. Communication Configuration for Workstation with OMR

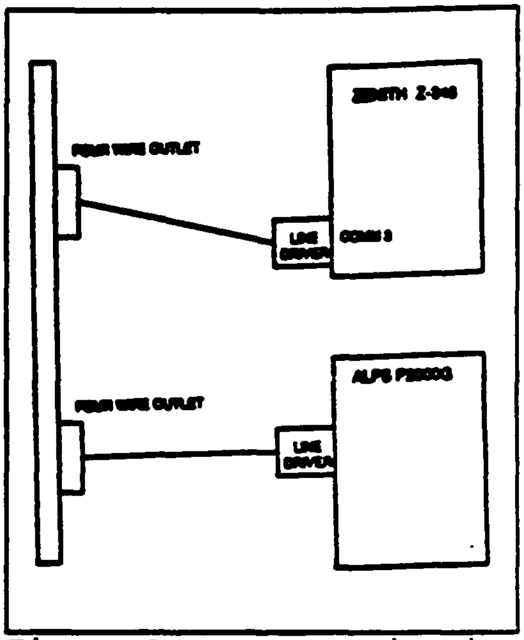

Figure A-6. Communication Configuration for Workstation without OMR

At Bergstrom AFB, base communications provides a four wire outlet in the designated work area. A four wire line is run from the outlet to a line driver that is plugged into the appropriate device. All communications lines require two line drivers; one at the start and one at the end. Figure A-5 shows the basic work center communications configuration with an OMR while Figure A-6 shows a work center without an OMR.

The 56K line from Brooks AFB to Ellington is connected to 992NP multiplexer via a 56K modem located at Building 1382. The intrabase communications network at Ellington ANGB is comprised of a central multiplexer, located in Building 1382, which is connected to the work centers in other buildings through Gandalf MLDS-122 Short Haul Modems and RS232 twisted pair cables. Work centers located in building 1382 are connected to the multiplexer directly by RS232 cables. The Ellington ANGB communications network architecture is shown in Figure A-7.

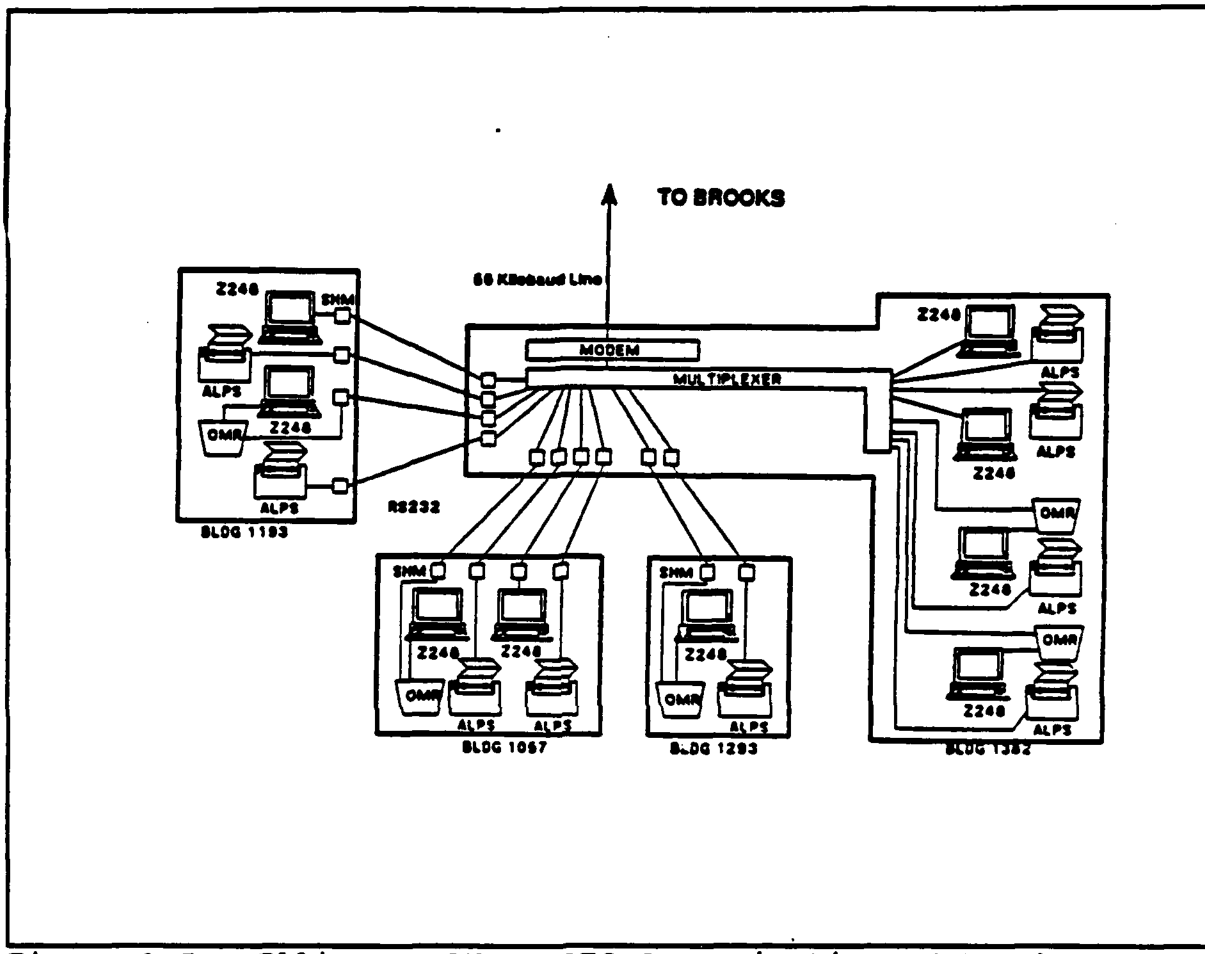

Figure A-7. Ellington ANGB AOTS Communications Network

B. Software Description.

1. Architecture. The AOTS software is not designed according to a strict hierarchy. Instead, it consists of a large number of linked software units. Figure A-8 illustrates the AOTS software architecture.

An Ada "unit" is the smallest collection of lines that an Ada compiler can compile. Units can be procedures, functions, generic templates, or packages. In addition, packages can be made up of specification and body units. VAX Ada uses the concept of a "closure" to describe the complete set of compiled units that must be linked to execute a program. For an Ada program to be executed, each unit in the closure must be compiled and linked. The closure includes all Ada units listed in the context statements (WITH and USE) contained in the top level procedure unit in the program. The closure also includes all units that are included in the context statements of the called units. This tiering continues until the compiler includes all units referenced in all context statements in the program. The closure may also include library modules written in other languages. These "foreign" modules are connected through the VAX Ada import and export compiler instructions.

The AOTS prototype includes many units which were written for the ISS. Some of these units are written in Ada, others are modules written in FORTRAN and MACRO. In order to describe the AOTS architecture, it is necessary to define what units comprise the AOTS. The prototype AOTS described in this document is AOTS Version 2.1 which was released September 1988. Table A-6 presents the AOTS Version Description Document which lists the units, by CMS name, that comprised the AOTS Version 2.1. In addition to the units required for the AOTS prototype main program, units that provide tools to maintain, document, and update files in the prototype system are included in this list.

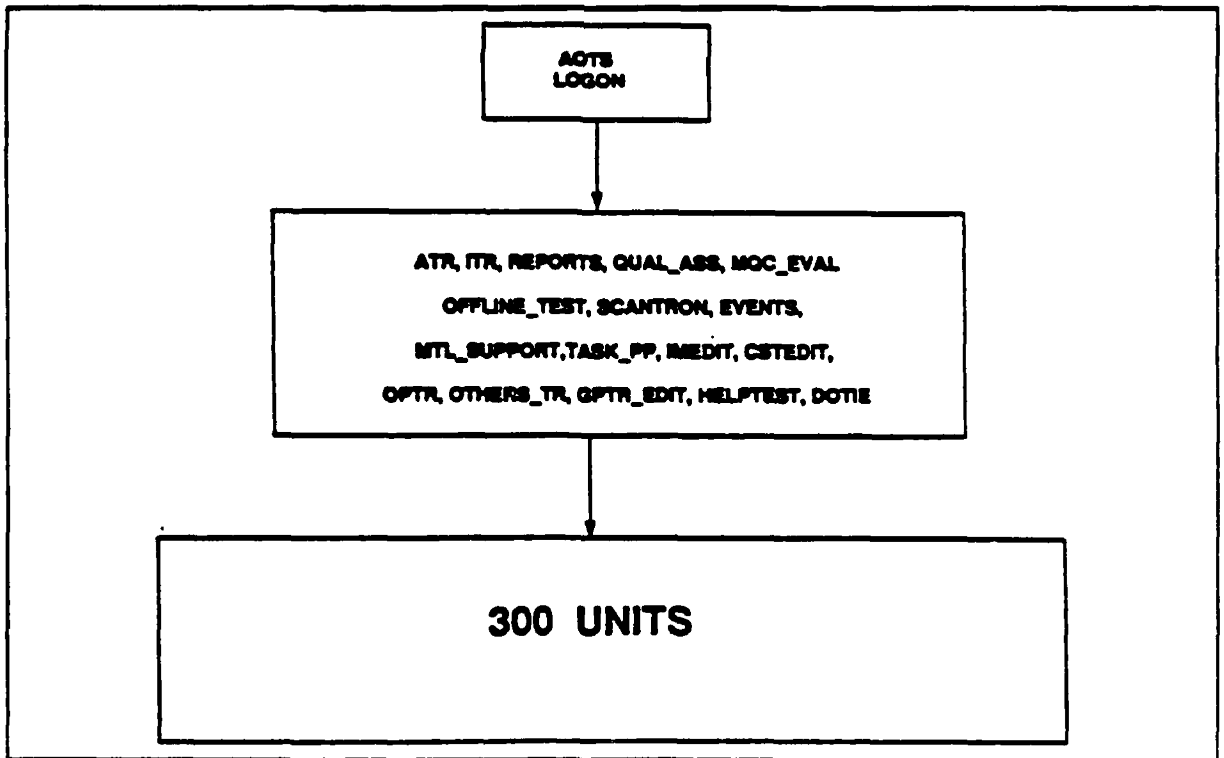

Figure A-8. AOTS Software Architecture

Table A-6. AOTS Version Description Document 2.1

Element generations in DEC/CMS Library DISK$AOTS2:[TEST.SE.AOTS.CMS]

A01USRMAN.ADA	3	21-JUN-1988	14:44:07	KB09	"RELEASE 1.9"
A01USRMAN_.ADA	3	21-JUN-1988	14:44:29	KB09	"RELEASE 1.9"
ACCESSB.ADA	1	16-MAY-1987	12:02:10	KB06	"Initial Beta release"
ACCESSS.ADA	1	16-MAY-1987	12:02:12	KB06	"Initial Beta release"
AD_NOC.COM	3	17-AUG-1988	17:05:55	KB09	"RELEASE 2.1"
AD_NOC.SAS	2	12-JUL-1988	17:05:48	KB09	"RELEASE 2.0"
AD_NOC.SFS	3	17-AUG-1988	17:06:35	KB09	"RELEASE 2.1"
CA1CRA.ADA	4	18-MAY-1988	15:40:00	KB09	"RELEASE 1.8"
CA1CRA_.ADA	3	21-JUN-1988	14:44:51	KB09	"RELEASE 1.9"
CABGEN.ADA	4	21-JUN-1988	14:45:15	KB09	"RELEASE 1.9"
CABGEN_.ADA	4	21-JUN-1988	14:45:39	KB09	"RELEASE 1.9"
CACR.ADA	3	21-JUN-1988	14:46:07	KB09	"RELEASE 1.9"
CACR_.ADA	3	21-JUN-1988	14:46:38	KB09	"RELEASE 1.9"
CAGEN.ADA	3	21-JUN-1988	14:47:17	KB09	"RELEASE 1.9"
CAGEN_.ADA	3	21-JUN-1988	14:47:49	KB09	"RELEASE 1.9"
CTCR_.ADA	3	21-JUN-1988	14:48:14	KB09	"RELEASE 1.9"
CTEXP_.ADA	3	21-JUN-1988	14:50:52	KB09	"RELEASE 1.9"
DEV-SELIST.DAT	2	21-JUN-1988	16:22:31	KB09	"RELEASE 1.9"
EVL001B.ADA	32	17-AUG-1988	17:07:32	KB09	"RELEASE 2.1"
EVL001S.ADA	13	17-AUG-1988	17:08:17	KB09	"RELEASE 2.1"
EVL002B.ADA	3	17-AUG-1988	17:08:42	KB09	"RELEASE 2.1"
EVL002S.ADA	1	16-MAY-1987	11:59:17	KB06	"Initial Beta release"
EVL003S.ADA	6	18-MAY-1988	15:42:42	KB09	"RELEASE 1.8"
EVL005B.ADA	18	17-AUG-1988	17:09:15	KB09	"RELEASE 2.1"
EVL005S.ADA	7	19-MAY-1988	19:02:13	KB06	"RELEASE 1.8 - Test Editor Fixes"
EVL006B.ADA	3	21-JUN-1988	14:58:23	KB09	"RELEASE 1.9"
EVL006S.ADA	8	12-JUL-1988	17:10:04	KB09	"RELEASE 2.0"
EVL007B.ADA	22	17-AUG-1988	17:10:07	KB09	"RELEASE 2.1"
EVL007S.ADA	10	21-JUN-1988	14:59:59	KB09	"RELEASE 1.9"
EVL008B.ADA	20	17-AUG-1988	17:11:13	KB09	"RELEASE 2.1"
EVL008S.ADA	7	18-MAY-1988	15:46:28	KB09	"RELEASE 1.8"
EVL009B.ADA	14	17-AUG-1988	17:12:41	KB09	"RELEASE 2.1"
EVL009S.ADA	5	18-MAY-1988	15:47:27	KB09	"RELEASE 1.8"
EVL010B.ADA	1	16-MAY-1987	11:59:49	KB06	"Initial Beta release"
EVL010S.ADA	1	16-MAY-1987	11:59:51	KB06	"Initial Beta release"
EVL011B.ADA	1	16-MAY-1987	11:59:53	KB06	"Initial Beta release"
EVL011S.ADA	1	16-MAY-1987	11:59:55	KB06	"Initial Beta release"
EVL012B.ADA	4	10-MAR-1988	10:26:33	KB09	"RELEASE 1.7"
EVL012S.ADA	3	5-OCT-1987	17:14:14	KB06	"Added support for graphic domain"
EVL013S.ADA	1	16-MAY-1987	12:00:00	KB06	"Initial Beta release"
EVL014B.ADA	8	5-OCT-1987	17:16:01	KB06	"Added delete flag and code to support it"
EVL014S.ADA	4	5-OCT-1987	17:15:30	KB06	"Added delete flag and code to support it"
EVL015B.ADA	5	5-OCT-1987	17:17:09	KB06	"Added delete flag and code to support it"
EVL015S.ADA	4	5-OCT-1987	17:16:36	KB06	"Added delete flag and code to support it"
EVL016B.ADA	6	10-MAR-1988	10:26:47	KB09	"RELEASE 1.7"
EVL016S.ADA	2	21-SEP-1987	15:29:11	KB09	"Version 1.5"
EVL017B.ADA	8	17-AUG-1988	17:13:35	KB09	"RELEASE 2.1"
EVL017S.ADA	2	21-JUN-1988	15:02:05	KB09	"RELEASE 1.9"
EVL018B.ADA	7	17-AUG-1988	17:14:28	KB09	"RELEASE 2.1"
EVL018S.ADA	2	21-JUN-1988	15:12:36	KB09	"RELEASE 1.9"
EVL019B.ADA	2	10-MAR-1988	10:27:04	KB09	"RELEASE 1.7"
EVL019S.ADA	3	18-MAY-1988	15:47:48	KB09	"RELEASE 1.8"
EVL020S.ADA	2	21-JUN-1988	15:12:47	KB09	"RELEASE 1.9"
EVL021B.ADA	2	21-JUN-1988	15:13:03	KB09	"RELEASE 1.9"
EVL021S.ADA	2	21-JUN-1988	15:13:16	KB09	"RELEASE 1.9"
EVL022B.ADA	3	12-JUL-1988	17:11:56	KB09	"RELEASE 2.0"
EVL022S.ADA	2	21-JUN-1988	15:13:44	KB09	"RELEASE 1.9"
EVL023P.ADA	2	21-JUN-1988	15:13:57	KB09	"RELEASE 1.9"
EVL024P.ADA	6	12-JUL-1988	17:12:12	KB09	"RELEASE 2.0"

Table A-6. AOTS Version Description Document 2.1 (Continued)

GENEDITB.ADA	2	16-NOV-1987	09:39:01	KB09	"Release 1.6"
GENEDITS.ADA	2	16-NOV-1987	09:39:36	KB09	"Release 1.6"
GENEDITT.ADA	1	26-OCT-1987	11:54:45	KB09	""
GRLIB.ADA	4	21-JUN-1988	15:14:15	KB09	"RELEASE 1.9"
GRLIB_.ADA	3	21-JUN-1988	15:14:31	KB09	"RELEASE 1.9"
GRTYPES.ADA	2	18-MAY-1988	15:49:29	KB09	"RELEASE 1.8"
GRTYPES_.ADA	3	21-JUN-1988	15:14:48	KB09	"RELEASE 1.9"
ITEM_EDITOR.ADA	1	27-AUG-1987	14:35:27	KB06	"design body for multi-page screen item editting"
ITEM_EDITOR_.ADA	1	27-AUG-1987	14:34:56	KB06	"design spec for multi-page screen item editting"
ITEM_SUP.ADA	2	16-NOV-1987	10:26:55	KB09	"Release 1.6"
LGLIB.ADA	3	21-JUN-1988	15:15:05	KB09	"RELEASE 1.9"
LGLIB_.ADA	3	21-JUN-1988	15:15:17	KB09	"RELEASE 1.9"
LGTYP_.ADA	3	21-JUN-1988	15:15:34	KB09	"RELEASE 1.9"
MGT001P.ADA	3	18-MAY-1988	15:51:25	KB09	"RELEASE 1.8"
MGT002B.ADA	33	17-AUG-1988	17:16:00	KB09	"RELEASE 2.1"
MGT002S.ADA	9	17-AUG-1988	17:17:20	KB09	"RELEASE 2.1"
MGT003B.ADA	39	2-SEP-1988	08:32:21	KB09	"Re-fix of task validation"
MGT003S.ADA	9	17-AUG-1988	17:20:30	KB09	"RELEASE 2.1"
MGT004B.ADA	11	17-AUG-1988	17:21:08	KB09	"RELEASE 2.1"
MGT004S.ADA	5	17-AUG-1988	17:21:39	KB09	"RELEASE 2.1"
MGT005B.ADA	2	18-MAY-1988	15:59:08	KB09	"RELEASE 1.8"
MGT005S.ADA	1	16-MAY-1987	11:51:45	KB06	"Initial Beta release"
MGT006B.ADA	8	16-NOV-1987	09:48:23	KB09	"Release 1.6"
MGT006S.ADA	2	4-JUN-1987	12:12:52	KB09	"Release 1.2"
MGT007B.ADA	22	17-AUG-1988	17:22:25	KB09	"RELEASE 2.1"
MGT007S.ADA	8	18-MAY-1988	16:01:22	KB09	"RELEASE 1.8"
MGT008B.ADA	7	16-NOV-1987	09:49:58	KB09	"Release 1.6"
MGT008S.ADA	4	21-SEP-1987	18:54:56	KB09	"Version 1.5"
MGT009P.ADA	5	17-AUG-1988	17:23:27	KB09	"RELEASE 2.1"
MGT010B.ADA	1	16-MAY-1987	11:52:08	KB06	"Initial Beta release"
MGT010S.ADA	1	16-MAY-1987	11:52:10	KB06	"Initial Beta release"
MGT013B.ADA	4	4-DEC-1987	15:05:36	KB09	""
MGT013S.ADA	1	16-MAY-1987	11:52:15	KB06	"Initial Beta release"
MGT014B.ADA	1	16-MAY-1987	11:52:17	KB06	"Initial Beta release"
MGT014S.ADA	1	16-MAY-1987	11:52:19	KB06	"Initial Beta release"
MGT015B.ADA	9	17-AUG-1988	17:23:53	KB09	"RELEASE 2.1"
MGT015S.ADA	5	12-JUL-1988	17:19:47	KB09	"RELEASE 2.0"
MGT016B.ADA	22	17-AUG-1988	17:24:19	KB09	"RELEASE 2.1"
MGT016S.ADA	6	12-JUL-1988	17:21:07	KB09	"RELEASE 2.0"
MGT017B.ADA	17	17-AUG-1988	17:24:49	KB09	"RELEASE 2.1"
MGT017S.ADA	5	17-AUG-1988	17:25:10	KB09	"RELEASE 2.1"
MGT018B.ADA	34	17-AUG-1988	17:25:45	KB09	"RELEASE 2.1"
MGT018S.ADA	12	12-JUL-1988	17:23:31	KB09	"RELEASE 2.0"
MGT019P.ADA	1	16-MAY-1987	11:52:43	KB06	"Initial Beta release"
MGT020P.ADA	2	14-AUG-1987	10:24:56	KB09	"Release 1.4"
MGT023B.ADA	17	17-AUG-1988	17:26:24	KB09	"RELEASE 2.1"
MGT023S.ADA	4	12-JUL-1988	17:24:53	KB09	"RELEASE 2.0"
MGT024B.ADA	16	31-AUG-1988	08:41:44	KB06	"Patch to reduce impact of final MTL installation"
MGT024S.ADA	6	18-MAY-1988	16:10:16	KB09	"RELEASE 1.8"
MGT025P.ADA	2	18-MAY-1988	16:10:35	KB09	"RELEASE 1.8"
MGT026P.ADA	2	16-NOV-1987	09:57:11	KB09	"Release 1.6"
MGT027P.ADA	1	4-JUN-1987	14:10:59	KB09	"Initial Beta Release"
MGT028P.ADA	1	19-MAY-1987	09:17:51	KB06	""
MGT029P.ADA	1	16-MAY-1987	11:52:58	KB06	"Initial Beta release"
MGT030B.ADA	18	23-AUG-1988	15:23:39	KB06	"Init of group event fields"
MGT030S.ADA	3	21-JUN-1988	15:25:10	KB09	"RELEASE 1.9"
MGT031B.ADA	15	25-AUG-1988	16:13:37	KB06	"Cleanup of QC_Eval and Daily_Run"
MGT031S.ADA	3	12-JUL-1988	17:27:51	KB09	"RELEASE 2.0"
MGT032B.ADA	24	26-AUG-1988	15:05:36	KB17	""
MGT032S.ADA	5	25-AUG-1988	17:01:58	KB06	""
MGT033B.ADA	14	6-SEP-1988	06:39:05	KB09	"Prevent user from updating own training history"

Table A-6. AOTS Version Description Document 2.1 (Continued)

MGT033S.ADA	7	17-AUG-1988	17:29:14	KB09	"RELEASE 2.1"
MGT034B.ADA	2	18-MAY-1988	16:15:52	KB09	"RELEASE 1.8"
MGT034S.ADA	2	18-MAY-1988	16:18:16	KB09	"RELEASE 1.8"
MGT035S.ADA	2	18-MAY-1988	16:18:44	KB09	"RELEASE 1.8"
MGT036P.ADA	2	18-MAY-1988	16:19:05	KB09	"RELEASE 1.8"
MGT037B.ADA	2	18-MAY-1988	16:19:24	KB09	"RELEASE 1.8"
MGT037S.ADA	2	18-MAY-1988	16:19:41	KB09	"RELEASE 1.8"
MGT038B.ADA	6	31-AUG-1988	09:11:37	KB06	"Correction to allow scheduling of PME and OFT"
MGT038S.ADA	2	21-JUN-1988	15:30:28	KB09	"RELEASE 1.9"
MGT039B.ADA	6	12-JUL-1988	17:31:03	KB09	"RELEASE 2.0"
MGT039S.ADA	2	18-MAY-1988	16:20:27	KB09	"RELEASE 1.8"
MGT040B.ADA	6	17-AUG-1988	17:29:55	KB09	"RELEASE 2.1"
MGT040S.ADA	3	21-JUN-1988	15:31:45	KB09	"RELEASE 1.9"
MGT042B.ADA	9	22-AUG-1988	12:51:17	KB11	""
MGT042S.ADA	2	18-MAY-1988	16:21:57	KB09	"RELEASE 1.8"
MGT043B.ADA	2	18-MAY-1988	16:22:20	KB09	"RELEASE 1.8"
MGT043S.ADA	2	18-MAY-1988	16:22:39	KB09	"RELEASE 1.8"
MGT045B.ADA	4	12-JUL-1988	17:32:54	KB09	"RELEASE 2.0"
MGT045S.ADA	3	21-JUN-1988	15:32:43	KB09	"RELEASE 1.9"
MGT046B.ADA	4	12-JUL-1988	17:33:25	KB09	"RELEASE 2.0"
MGT046S.ADA	3	21-JUN-1988	15:33:24	KB09	"RELEASE 1.9"
MGT047B.ADA	4	12-JUL-1988	17:33:57	KB09	"RELEASE 2.0"
MGT047S.ADA	3	21-JUN-1988	15:34:07	KB09	"RELEASE 1.9"
MGT048B.ADA	3	17-AUG-1988	17:30:46	KB09	"RELEASE 2.1"
MGT048S.ADA	2	21-JUN-1988	15:37:22	KB09	"RELEASE 1.9"
MGT049B.ADA	5	17-AUG-1988	17:31:08	KB09	"RELEASE 2.1"
MGT049S.ADA	2	18-MAY-1988	16:25:07	KB09	"RELEASE 1.8"
MGT050B.ADA	5	12-JUL-1988	17:35:02	KB09	"RELEASE 2.0"
MGT050S.ADA	2	18-MAY-1988	16:25:42	KB09	"RELEASE 1.8"
MGT051B.ADA	3	12-JUL-1988	17:35:29	KB09	"RELEASE 2.0"
MGT051S.ADA	2	18-MAY-1988	16:26:16	KB09	"RELEASE 1.8"
MGT052B.ADA	2	18-MAY-1988	16:26:33	KB09	"RELEASE 1.8"
MGT052S.ADA	2	18-MAY-1988	16:26:55	KB09	"RELEASE 1.8"
MGT054B.ADA	18	6-SEP-1988	06:37:44	KB09	"Prevent user from certifying self"
MGT054S.ADA	5	26-AUG-1988	15:27:59	KB06	"Moved Install ITR update code to Update_training_Records"
MGT055B.ADA	8	17-AUG-1988	17:32:45	KB09	"RELEASE 2.1"
MGT055S.ADA	4	17-AUG-1988	17:33:14	KB09	"RELEASE 2.1"
MGT056B.ADA	2	18-MAY-1988	16:27:17	KB09	"RELEASE 1.8"
MGT056S.ADA	2	18-MAY-1988	16:27:36	KB09	"RELEASE 1.8"
MGT057P.ADA	5	31-AUG-1988	08:44:05	KB06	"Patch to reduce impact of final MTL installation"
MGT058S.ADA	4	17-AUG-1988	17:34:02	KB09	"RELEASE 2.1"
MGT059S.ADA	3	17-AUG-1988	17:34:30	KB09	"RELEASE 2.1"
MGT060B.ADA	7	17-AUG-1988	17:35:05	KB09	"RELEASE 2.1"
MGT060S.ADA	4	17-AUG-1988	17:37:41	KB09	"RELEASE 2.1"
MGT060T.ADA	2	17-AUG-1988	17:35:39	KB09	"RELEASE 2.1"
MGT060TA.ADA	2	17-AUG-1988	17:35:55	KB09	"RELEASE 2.1"
MGT060TB.ADA	2	17-AUG-1988	17:36:19	KB09	"RELEASE 2.1"
MGT060TD.ADA	2	17-AUG-1988	17:36:40	KB09	"RELEASE 2.1"
MGT061P.ADA	4	17-AUG-1988	17:37:13	KB09	"RELEASE 2.1"
MGT062B.ADA	8	31-AUG-1988	08:43:42	KB06	"Patch to reduce impact of final MTL installation"
MGT062S.ADA	6	17-AUG-1988	17:38:20	KB09	"RELEASE 2.1"
MGT063P.ADA	7	17-AUG-1988	17:38:41	KB09	"RELEASE 2.1"
MGT064B.ADA	6	17-AUG-1988	17:39:02	KB09	"RELEASE 2.1"
MGT064S.ADA	3	12-JUL-1988	17:42:31	KB09	"RELEASE 2.0"
MGT065B.ADA	11	22-AUG-1988	12:49:11	KB11	""
MGT065S.ADA	11	22-AUG-1988	12:49:59	KB11	""
MGT066P.ADA	6	17-AUG-1988	17:40:06	KB09	"RELEASE 2.1"
MGT067P.ADA	5	17-AUG-1988	17:40:24	KB09	"RELEASE 2.1"
MGT068C.COM	4	17-AUG-1988	17:40:43	KB09	"RELEASE 2.1"
MGT070X.SAS	3	17-AUG-1988	17:40:59	KB09	"RELEASE 2.1"
MGT071X.SAS	4	17-AUG-1988	17:41:16	KB09	"RELEASE 2.1"

Table A-6. AOTS Version Description Document 2.1 (Continued)

MGT072X.SAS	4	17-AUG-1988	17:41:35	KB09	"RELEASE 2.1"
MGT073X.SAS	3	17-AUG-1988	17:41:52	KB09	"RELEASE 2.1"
MGT074X.SAS	3	17-AUG-1988	17:42:08	KB09	"RELEASE 2.1"
MGT075X.SAS	3	17-AUG-1988	17:42:25	KB09	"RELEASE 2.1"
MGT076X.SAS	3	17-AUG-1988	17:42:41	KB09	"RELEASE 2.1"
MGT077X.SAS	3	17-AUG-1988	17:42:58	KB09	"RELEASE 2.1"
MGT078X.SAS	2	21-JUN-1988	15:48:16	KB09	"RELEASE 1.9"
MGT079X.SAS	4	17-AUG-1988	17:43:17	KB09	"RELEASE 2.1"
MGT080X.SAS	2	21-JUN-1988	15:49:02	KB09	"RELEASE 1.9"
MGT081X.SAS	2	21-JUN-1988	15:49:21	KB09	"RELEASE 1.9"
MGT082X.SAS	3	17-AUG-1988	17:43:33	KB09	"RELEASE 2.1"
MGT083X.SAS	3	17-AUG-1988	17:43:49	KB09	"RELEASE 2.1"
MGT084X.SAS	4	17-AUG-1988	17:44:07	KB09	"RELEASE 2.1"
MGT085P.ADA	6	25-AUG-1988	16:14:55	KB06	"Cleanup of QC_Eval and Daily_Run"
MGT086B.ADA	4	12-JUL-1988	17:46:20	KB09	"RELEASE 2.0"
MGT086S.ADA	2	18-MAY-1988	16:30:43	KB09	"RELEASE 1.8"
MGT087B.ADA	6	17-AUG-1988	17:44:42	KB09	"RELEASE 2.1"
MGT087S.ADA	3	21-JUN-1988	15:51:55	KB09	"RELEASE 1.9"
MGT088B.ADA	4	17-AUG-1988	17:45:02	KB09	"RELEASE 2.1"
MGT088S.ADA	2	21-JUN-1988	15:52:39	KB09	"RELEASE 1.9"
MGT089S.ADA	3	17-AUG-1988	17:45:19	KB09	"RELEASE 2.1"
MGT090P.ADA	4	17-AUG-1988	17:45:35	KB09	"RELEASE 2.1"
MGT091S.ADA	2	18-MAY-1988	16:32:26	KB09	"RELEASE 1.8"
MGT092B.ADA	8	17-AUG-1988	17:45:53	KB09	"RELEASE 2.1"
MGT092S.ADA	2	18-MAY-1988	16:33:28	KB09	"RELEASE 1.8"
MGT093B.ADA	8	17-AUG-1988	17:46:15	KB09	"RELEASE 2.1"
MGT093S.ADA	2	18-MAY-1988	16:34:27	KB09	"RELEASE 1.8"
MGT094B.ADA	9	17-AUG-1988	17:46:38	KB09	"RELEASE 2.1"
MGT094S.ADA	3	17-AUG-1988	17:46:57	KB09	"RELEASE 2.1"
MGT095B.ADA	5	17-AUG-1988	17:47:14	KB09	"RELEASE 2.1"
MGT095S.ADA	4	17-AUG-1988	17:47:30	KB09	"RELEASE 2.1"
MGT096B.ADA	3	21-JUN-1988	15:54:38	KB09	"RELEASE 1.9"
MGT096S.ADA	2	18-MAY-1988	16:36:50	KB09	"RELEASE 1.8"
MGT097B.ADA	3	17-AUG-1988	17:47:45	KB09	"RELEASE 2.1"
MGT097S.ADA	2	18-MAY-1988	16:37:32	KB09	"RELEASE 1.8"
MGT098B.ADA	11	17-AUG-1988	17:48:04	KB09	"RELEASE 2.1"
MGT098S.ADA	3	12-JUL-1988	17:50:09	KB09	"RELEASE 2.0"
MGT099S.ADA	3	21-JUN-1988	15:55:13	KB09	"RELEASE 1.9"
MGT100B.ADA	10	17-AUG-1988	17:48:21	KB09	"RELEASE 2.1"
MGT100S.ADA	3	21-JUN-1988	15:55:46	KB09	"RELEASE 1.9"
MGT101P.ADA	6	31-AUG-1988	08:44:37	KB06	"Patch to reduce impact of final NTL installation"
MGT102B.ADA	2	21-JUN-1988	15:56:18	KB09	"RELEASE 1.9"
MGT102S.ADA	2	21-JUN-1988	15:56:35	KB09.	"RELEASE 1.9"
MGT103B.ADA	2	21-JUN-1988	15:56:49	KB09	"RELEASE 1.9"
MGT103S.ADA	2	21-JUN-1988	15:57:01	KB09	"RELEASE 1.9"
MGT104C.COM	2	21-JUN-1988	15:57:16	KB09	"RELEASE 1.9"
MGT105C.COM	2	21-JUN-1988	15:57:27	KB09	"RELEASE 1.9"
MGT106C.COM	2	21-JUN-1988	15:57:40	KB09	"RELEASE 1.9"
MGT107C.COM	2	21-JUN-1988	15:57:52	KB09	"RELEASE 1.9"
MGT108C.COM	2	21-JUN-1988	15:58:06	KB09	"RELEASE 1.9"
MGT109C.COM	2	21-JUN-1988	15:58:18	KB09	"RELEASE 1.9"
MGT110C.COM	2	21-JUN-1988	15:58:30	KB09	"RELEASE 1.9"
MGT111C.COM	2	21-JUN-1988	15:58:46	KB09	"RELEASE 1.9"
MGT112B.ADA	6	31-AUG-1988	09:54:28	KB11	""
MGT112S.ADA	2	21-JUN-1988	15:59:16	KB09	"RELEASE 1.9"
MGT113S.ADA	4	17-AUG-1988	17:49:10	KB09	"RELEASE 2.1"
MGT114P.ADA	5	17-AUG-1988	17:52:34	KB09	"RELEASE 2.1"
MGT115P.ADA	5	22-AUG-1988	12:51:37	KB11	""
MGT116B.ADA	4	17-AUG-1988	17:53:09	KB09	"RELEASE 2.1"
MGT116S.ADA	2	21-JUN-1988	16:00:45	KB09	"RELEASE 1.9"
MGT117C.COM	2	21-JUN-1988	16:01:08	KB09	"RELEASE 1.9"

Figure A-6. AOTS Version Description Document 2.1 (Continued)

[illegible]T118P.ADA	5	22-AUG-1988 08:45:54	KB11	""
[illegible]T119X.BAS	2	17-AUG-1988 17:53:43	KB09	"RELEASE 2.1"
[illegible]T120P.ADA	3	22-AUG-1988 08:46:19	KB11	""
REPORTSP.ADA	2	18-MAY-1988 16:40:35	KB09	"RELEASE 1.8"
SACCESS.ADA	3	21-JUN-1988 16:01:26	KB09	"RELEASE 1.9"
SACCESS_.ADA	3	21-JUN-1988 16:01:47	KB09	"RELEASE 1.9"
SDOKEYTTP_.ADA	3	21-JUN-1988 16:02:15	KB09	"RELEASE 1.9"
SD3SR_.ADA	3	21-JUN-1988 16:02:39	KB09	"RELEASE 1.9"
SD4PASS_.ADA	3	21-JUN-1988 16:03:06	KB09	"RELEASE 1.9"
SDINTSTO_.ADA	2	21-JUN-1988 16:03:31	KB09	"RELEASE 1.9"
SELIST.DAT	3	17-AUG-1988 17:54:11	KB09	"RELEASE 2.1"
SUP001B.ADA	27	25-AUG-1988 17:10:34	KB06	"Mod to Field_Setup to limit display to max of F_Length"
SUP001S.ADA	12	17-AUG-1988 17:55:32	KB09	"RELEASE 2.1"
SUP004B.ADA	1	16-MAY-1987 12:00:42	KB06	"Initial Beta release"
SUP004S.ADA	1	16-MAY-1987 12:00:44	KB06	"Initial Beta release"
SUP005B.ADA	12	3-DEC-1987 13:39:15	KB06	""
SUP005S.ADA	3	26-OCT-1987 12:35:39	KB09	""
SUP006B.ADA	1	16-MAY-1987 12:00:52	KB06	"Initial Beta release"
SUP006S.ADA	1	16-MAY-1987 12:00:54	KB06	"Initial Beta release"
SUP007B.ADA	10	17-AUG-1988 17:55:56	KB09	"RELEASE 2.1"
SUP007S.ADA	5	17-AUG-1988 17:56:17	KB09	"RELEASE 2.1"
SUP008P.ADA	2	3-DEC-1987 13:39:35	KB06	""
SUP009B.ADA	11	21-JUN-1988 16:06:49	KB09	"RELEASE 1.9"
SUP009S.ADA	2	21-SEP-1987 15:35:21	KB09	"Version 1.5"
SUP010P.ADA	2	21-JUN-1988 16:07:15	KB09	"RELEASE 1.9"
SUP011P.ADA	2	21-JUN-1988 16:07:35	KB09	"RELEASE 1.9"
SUP012B.ADA	2	21-JUN-1988 16:07:57	KB09	"RELEASE 1.9"
SUP012S.ADA	2	21-JUN-1988 16:08:16	KB09	"RELEASE 1.9"
SUP013B.ADA	4	26-OCT-1987 11:38:30	KB09	""
SUP013S.ADA	2	26-OCT-1987 11:37:48	KB09	""
SUP014B.ADA	7	17-AUG-1988 17:56:39	KB09	"RELEASE 2.1"
SUP014S.ADA	2	17-AUG-1988 17:56:57	KB09	"RELEASE 2.1"
SUP014T.ADA	2	17-AUG-1988 17:57:12	KB09	"RELEASE 2.1"
SUP015B.ADA	2	21-JUN-1988 16:08:35	KB09	"RELEASE 1.9"
SUP015S.ADA	2	21-JUN-1988 16:08:55	KB09	"RELEASE 1.9"
SUP016B.ADA	2	21-JUN-1988 16:09:18	KB09	"RELEASE 1.9"
SUP016S.ADA	2	21-JUN-1988 16:09:43	KB09	"RELEASE 1.9"
SUP017B.ADA	2	21-JUN-1988 16:10:10	KB09	"RELEASE 1.9"
SUP017S.ADA	2	21-JUN-1988 16:10:37	KB09	"RELEASE 1.9"
SUP018A.ASM	2	21-JUN-1988 16:11:08	KB09	"RELEASE 1.9"
SUP019A.ASM	2	21-JUN-1988 16:11:36	KB09	"RELEASE 1.9"
SUP020P.ADA	2	21-JUN-1988 16:12:07	KB09	"RELEASE 1.9"
SUP021B.ADA	2	21-JUN-1988 16:12:31	KB09	"RELEASE 1.9"
SUP021S.ADA	2	21-JUN-1988 16:12:56	KB09	"RELEASE 1.9"
SUP022B.ADA	2	21-JUN-1988 16:13:21	KB09	"RELEASE 1.9"
SUP022S.ADA	2	21-JUN-1988 16:13:46	KB09	"RELEASE 1.9"
SUP023B.ADA	2	21-JUN-1988 16:14:11	KB09	"RELEASE 1.9"
SUP023S.ADA	2	21-JUN-1988 16:14:34	KB09	"RELEASE 1.9"
SUP024B.ADA	3	17-AUG-1988 17:57:28	KB09	"RELEASE 2.1"
SUP024S.ADA	2	18-MAY-1988 16:52:30	KB09	"RELEASE 1.8"
SUP026B.ADA	23	17-AUG-1988 17:57:52	KB09	"RELEASE 2.1"
SUP026S.ADA	5	12-JUL-1988 17:57:47	KB09	"RELEASE 2.0"
SUP027P.ADA	13	17-AUG-1988 17:58:19	KB09	"RELEASE 2.1"
SUP028B.ADA	9	17-AUG-1988 17:58:45	KB09	"RELEASE 2.1"
SUP028S.ADA	4	18-MAY-1988 16:56:31	KB09	"RELEASE 1.8"
SUP029B.ADA	3	21-JUN-1988 16:18:06	KB09	"RELEASE 1.9"
SUP029S.ADA	2	18-MAY-1988 16:57:35	KB09	"RELEASE 1.8"
SUP030S.ADA	8	17-AUG-1988 17:59:06	KB09	"RELEASE 2.1"
SUP031B.ADA	10	17-AUG-1988 17:59:25	KB09	"RELEASE 2.1"
SUP031S.ADA	4	21-JUN-1988 16:19:48	KB09	"RELEASE 1.9"
SUP032B.ADA	2	21-JUN-1988 16:20:17	KB09	"RELEASE 1.9"
SUP032S.ADA	2	21-JUN-1988 16:20:36	KB09	"RELEASE 1.9"
SUP033B.ADA	2	18-MAY-1988 17:01:20	KB09	"RELEASE 1.8"
SUP034P.ADA	2	21-JUN-1988 16:20:55	KB09	"RELEASE 1.9"
TBDTYPES_.ADA	4	21-JUN-1988 16:21:10	KB09	"RELEASE 1.9"

(a) Prototype Main Program. Each Ada program must contain at least one subprogram. The AOTS prototype contains 29 procedures that qualify as subprograms. The subprograms are usually identified with a "P" (for procedure) suffix to the configuration management number in the Version Description Document.

The main program for the AOTS prototype is the procedure AOTS LOGON, and its execution creates the interactive environment for courseware developers, trainees, and supervisors. The AOTS prototype main program includes the unit AOTS LOGON and all units in its closure. The closure was determined by creating a linked image of AOTS LOGON. This was accomplished by creating an Ada library, and then entering into it all the units that are necessary to link the prototype. The VAX ACS ENTER UNIT command establishes pointers to the owning library. Then once the closure is established, the library directory contains a list of the units included in the closure. While any one unit can be compiled separately, the VAX Ada link command must name the subprogram unit.

The AOTS LOGON program consists of 309 units and modules. Of these, nine units are standard VAX Ada files. These nine are the specifications and bodies to CONDITION HANDLING, IO EXCEPTIONS, CALENDAR, STARLET, SEQUENTIAL IO, and TEXT IO. Forty-five modules are written in FORTRAN and eight are written in VAX/VMS Macro assembly language. The remaining 247 are Ada source code, and consist of both those originally written for ISS and those written specifically for AOTS prototype. Regardless of their origin, they all reside in the AOTS prototype libraries and are required for an image capable of execution. The file space required to store the source code for AOTS LOGON unit and modules is 3,012,739 bytes. Only the software maintainer's machine is required to store these files.

(b) Developed Software Components. Ada VAX ACS link commands must specify a subprogram unit. The subprogram unit provides a sequential list of steps and typically makes numerous calls to package units. Procedure and function units, then, are subprograms and were considered software components for this study. Table A-7 lists the Ada subprograms resident in the AOTS prototype CMS libraries. Closure for most of the 29 components was determined using the methodology described above for the main program.

The closures on some of these units were large and similar to each other in size. Of these large closures, some were included in the AOTS LOGON executable image, and other large and small components are tools and background programs that serve other functions. The following discussion is organized into three areas:

Table A-7. AOTS CMS Libraries Subprograms

AOTS_LOGON	BURSTER	CALL_DCL
CHECK_CHANGES	CSTEDIT	DAILY_RUN
EXTRACT	FINAL_MTL	FINAL_MTL_GEN
HELPER	IMEDIT	IMPLEMENT_MTL
ITR_BG	MAPEDIT	MTL_EDITOR
MTL_FUNCTION_SELECT	OPTR_EDITOR	OTR_EDITOR
PATH_OPEN	PDS_TO_AOTS	PRINT_REPORTS
REPORTS	SCANTRON	SCREEN_DUMP
TENATIVE_MTL	TERMINAL	TEST_PRINT
TPP_EDITOR	UPDATE_TRAINING_RECORDS	

1) large units included in the AOTS LOGON, 2) large units not included in AOTS LOGON, and 3) small units.

Seventeen of the components on the list above contain almost the same large number of units to become executable. Further, the names of the units in each closure are almost identical. These large components are CHECK CHANGES, CSTEDIT, DAILY RUN, EXTRACT, FINAL MTL GEN, IMEDIT, ITR BG, MTL EDITOR, MTL FUNCTION SELECT, OTR EDITOR, PDS TO AOTS, PRINT REPORTS, REPORTS, SCANTRON, TEST PRINT, TPP EDITOR, and UPDATE TRAINING RECORD.

(1) AOTS LOGON Components. Six of these large components are in the AOTS LOGON linker map. The 6 are CHECK CHANGES, CSTEDIT, IMEDIT, MTL FUNCTION SELECT, REPORTS, AND SCANTRON. Typical of these 6 is IMEDIT. When it is broken out and linked separately, it requires about 175 other units to form its closure. Realizing that approximately 300 units are in the AOTS prototype executable image, IMEDIT requires over 50% of them to become its own executable image. The software design rules for the AOTS prototype allow each procedure in the hierarchy to call all other units, so these units are very interdependent.

Another typical example is CSTEDIT. This procedure edits any Common Subtask for any of the AFSs defined in the AOTS system. It provides the ability to add, display, edit, print, and delete common tasks and information contained within the tasks specifications. This unit contains 22 comments and 27 commands in 81 total lines of source code. Other units in its context statements include AFS, SUBTASKS, SUB TYPES, ITEM, MTL BASIC, and TASK PP. The unit makes two calls to other units, operates on 13 files, and uses five global definitions. Comment statements consist of the obligatory minimum required by the software standards manual.

(2) Large Developed Software Tools. The remaining 11 large components are DAILY RUN, EXTRACT, FINAL MTL GEN, ITR BG, MTL EDITOR, OTR EDITOR, PDS TO AOTS, PRINT REPORTS, TEST PRINT, TPP EDITOR, and UPDATE TRAINING RECORD. These components are not in the AOTS LOGON link map, and perform other functions. Many of these names are recognized as major developed software tools such as PRINT REPORTS, that perform maintenance, documentation, and utility functions as their names imply.

(3) Other Developed Tools. In addition to the 17 major components, 11 components of various smaller sizes are contained in the CMS library. These components are: BURSTER, CALL DCL, HELPER, IMPLEMENT MTL, MAPEDIT, OPTR EDITOR, OTR EDITOR, PATH OPEN, SCREEN DUMP, TENTATIVE MTL, and TERMINAL. These components are also largely software tools, and a number of them are currently not in use.

2. Characteristics.

(a) User Interface. The AOTS software employs a menu-driven user interface. Menu-driven interfaces are typically easy to learn and therefore helpful to first time or infrequent users of a system. Unless they have an optional command mode, however, menu-driven systems become tedious for expert users to interact with. In spite of the fact that it is menu-driven, the AOTS user interface is not ideally designed for either first-time or experienced users. The AOTS prototype is not hierarchically designed, therefore a user can get to a single point in the software via many different paths. This is an advantage for the experienced user, but is confusing to a new user. The AOTS does not provide a command mode to aid the experienced user.

(b) Documentation. AOTS documentation is physically separated into volumes. Primary documents include the AOTS System Specification (CI# 7000), the Prime Item Specifications, the Computer Program Configuration Item Development Specifications, and the Computer Program Configuration Item Product Specifications

There are five Prime Item Specifications: 1) Management Subsystem, CI# 7100; 2) Training Development and Delivery Subsystem, CI# 7200; 3) Evaluation Subsystem, CI# 7300; 4) Computer Support Subsystem, CI# 7400; and 5) Personnel and Logistic Support Subsystem, CI# 7500.

There are four Computer Program Configuration Item Development Specifications: 1) Management CPCI, CI# 7411; 2)

Training Development and Delivery CPCI, CI# 7412; 3) Evaluation CPCI, CI# 7413; and 4) System Support CPCI, CI# 7414.

There are three Computer Program Configuration Item (CPCI) Product Specifications: 1) Management CPCI, CI# 7411 Volume II; 2) Evaluation CPCI, CI# 7413 Volume II; and 3) System Support CPCI, CI# 7414 Volume II.

The AOTS documentation has a number of shortcomings. The AOTS code relies on the self-documenting characteristics of Ada for data item descriptions, types, ranges, sizes, formats, sources, etc. No separate documentation is provided on how program input or output is accomplished, nor is any available on program error processing. In addition, the design specifications do not provide enough detail to trace elements as identifiable parts of the software from the system specification to the detailed design specification.

Coding standards for the AOTS software are contained in "Computer Programming Standards Document for Advanced On-the-Job Training System", 15 September 1986., "The Configuration Management Plan (CMP) for the Advanced On-the-Job Training System (AOTS)", 19 December 1986, and the "Advanced On-the-Job Training System Software Development Plan", 7 April 1986.

3. Metrics. In an attempt to determine the size and complexity of the prototype AOTS system, some characteristics of the software were classified and counted. The counting of these metrics was accomplished over a period of six months in support of a partitioning analysis of the current system and a cost analysis of a future AOTS system.

The prototype AOTS is comprised of approximately 320 units or files. Of these 320 units, 252 units were counted. Table A-8 lists the AOTS units that were counted and their metrics. There are 16 SAS units that are part of AOTS. The metrics for these 16 SAS units have been combined and are listed under the unit name "SAS PROGRAMS".

Table A-8 lists, from left to right, the CMS Number, Unit Name, programming language, total number of lines (carriage returns), number of comment lines, Source Lines of Code (SLOC), object code file size, the count of mix elements, the number of units called, and the number of units that call the unit.

Table A-8. AOTS File Metric List

CMS NUMBER	UNIT NAME	LANGUAGE	TOTAL LINES	COMMENT LINES	SLOC	OBJECT CODE FILE SIZE (BYTES)	MIX ELEMENTS MOPR	MINT	MREA	MONL	MDAT	MSTR	MMAT	NUMBER OF UNITS CALLED	NUMBER OF UNITS CALLED BY
EVL001B	BOE	ADA	3844	523	1647	192512	27	0	0	414	52	67	71	30	22
EVL001S	BOE_	ADA	594	404	109	10240	0	0	0	1	0	3	0	11	0
EVL002B	VERB_CHECK	ADA	401	64	198	19968	0	0	0	96	22	12	9	7	1
EVL002S	VERB_CHECK_	ADA	83	63	11	1536	0	0	0	1	1	0	0	2	0
EVL003S	OBJ_TYPES_	ADA	92	35	32	92672	0	0	0	2	0	2	0	5	41
EVL005B	TEST_EDITOR	ADA	1534	257	590	80896	0	0	0	131	30	29	20	20	3
EVL005S	TEST_EDITOR_	ADA	96	79	10	1536	0	0	0	1	0	0	0	4	0
EVL006B	TIB_TYPE_	ADA	375	155	112	16384	0	0	0	23	1	3	0	8	0
EVL006S	TIB_TYPE	ADA	270	69	116	17408	0	0	0	43	2	0	14	8	33
EVL007B	TEST_ITEM_BANK_EDITOR	ADA	1713	549	648	62464	8	3	0	121	32	7	15	20	6
EVL007S	TEST_ITEM_BANK_EDITOR_	ADA	261	216	31	2048	0	0	0	1	0	0	0	5	0
EVL008B	QUESTION	ADA	2093	493	908	87552	10	0	0	162	12	60	15	17	10
EVL008S	QUESTION_	ADA	372	249	81	8704	0	0	0	6	0	0	1	9	0
EVL009B	PM_OTG	ADA	1288	458	457	144896	15	1	0	85	15	24	11	13	8
EVL009S	PM_OTG_	ADA	242	167	50	5632	0	0	0	4	0	0	1	6	0
EVL012B	GRAFIX	ADA	413	105	138	17408	0	0	0	26	9	6	5	10	4
EVL012S	GRAFIX_	ADA	214	131	35	5632	0	0	0	4	0	1	0	2	0
EVL016B	CR_QUES	ADA	632	104	309	32768	4	0	0	52	31	20	1	13	4
EVL016S	CR_QUES_	ADA	198	125	39	4608	0	0	0	0	0	1	0	2	0
EVL017B	TEST_PRES	ADA	1661	400	787	70656	0	0	0	167	44	23	83	20	2
EVL017S	TEST_PRES_	ADA	171	79	64	7680	0	0	0	0	3	0	2	8	0
EVL018B	OFFLINE_TEST	ADA	1940	379	794	92672	0	0	0	290	90	59	48	26	8
EVL018S	OFFLINE_TEST_	ADA	268	160	78	8192	0	0	0	0	11	2	1	9	0
EVL019B	ALTS	ADA	1616	386	639	75776	0	2	0	123	16	16	17	12	9
EVL019S	ALTS_	ADA	330	197	90	11776	0	0	0	8	0	2	3	11	0
VL020S	FORM_DEFINITION_	ADA	46	23	14	2048	0	0	0	0	2	0	0	4	0
.VL021B	FORM_UTILITIES	ADA	1067	386	346	39424	2	0	0	66	0	68	20	7	1
EVL021S	FORM_UTILITIES_	ADA	207	121	51	6144	0	0	0	0	0	0	0	2	0
EVL022B	FORM_PROCESSING	ADA	625	155	243	23040	0	0	0	58	16	10	33	14	1
EVL022S	FORM_PROCESSING_	ADA	60	46	4	1536	0	0	0	0	0	0	0	1	0
EVL023P	SCANTRON	ADA	613	62	269	77824	3	2	0	50	5	12	14	19	1
EVL024P	TEST_PRINT	ADA	260	83	84	13312	3	0	0	25	7	3	0	10	0
EVL^25S	TIETYPES_	ADA	133	34	68	7680	0	0	0	0	0	0	0	4	0

Table A-8. AOTS File Metric List. (Continued)

CMS NUMBER	UNIT NAME	LANGUAGE	TOTAL LINES	COMMENT LINES	SLOC	OBJECT CODE FILE SIZE (BYTES)	MIX ELEMENTS MOPR	MINT	MREA	MONL	MDAT	MSTR	MMAT	NUMBER OF UNITS CALLED	NUMBER OF UNITS CALLED BY
EVL026P	PAATIE	ADA	2902	774	912	105984	3	0	0	65	0	33	99	13	0
EVL026P	TIE	ADA	2771	755	860	100864	3	0	0	63	0	33	95	14	22
EVL026P	SAVTIE	ADA	2887	774	912	105984	3	0	0	58	0	33	100	13	0
EVL027B	TIETIE	ADA	143	32	62	5120	0	0	0	4	4	0	1	12	1
EVL027S	TIETIE_	ADA	42	23	5	1024	0	0	0	0	0	0	0	1	0
EVL028B	DOTIE	ADA	515	75	162	27648	0	0	0	40	8	0	1	0	0
EVL028S	DOTIE_	ADA	44	24	5	1024	0	0	0	1	0	0	0	0	0
MGT002B	TASK_PP	ADA	5085	464	2388	226816	51	8	0	665	186	158	51	17	27
MGT002S	TASK_PP_	ADA	284	174	53	7168	0	0	0	11	0	0	0	5	0
MGT003B	MTL_SUPPORT	ADA	2693	230	1083	123904	26	9	0	191	85	20	29	32	4
MGT003S	MTL_SUPPORT_	ADA	44	35	5	1024	0	0	0	1	0	0	0	1	0
MGT004B	TASK_SELECT_PRIOR	ADA	747	70	309	91136	3	0	0	64	24	16	15	19	6
MGT004S	TASK_SELECT_PRIOR_	ADA	111	65	26	54784	0	0	0	1	0	0	1	5	0
MGT005B	CROSS_	ADA	206	119	29	4608	0	0	0	0	0	0	0	2	0
MGT005S	CROSS	ADA	501	120	157	14336	16	0	0	7	14	0	0	7	15
MGT007B	ITEM	ADA	4020	586	1749	150528	27	0	0	359	91	97	50	27	73
MGT007S	ITEM_	ADA	552	321	108	8704	0	0	0	4	0	0	0	6	48
MGT009P	IMEDIT	ADA	160	31	64	8704	0	0	0	10	0	1	1	10	1
MGT015B	AFSC	ADA	288	18	114	14848	2	0	0	32	2	3	0	9	66
MGT015S	AFSC_	ADA	197	115	43	7168	0	0	0	5	1	4	0	4	0
MGT016B	MTL_UNIT_CODES	ADA	2099	388	743	72704	23	0	0	179	27	118	26	15	14
MGT016S	MTL_UNIT_CODES_	ADA	260	153	50	9216	0	0	0	9	0	0	0	5	0
MGT017B	ACTIVITY	ADA	592	118	223	71168	2	0	0	65	18	7	9	15	12.
MGT017S	ACTIVITY_	ADA	118	74	24	3584	0	0	0	6	0	0	1	4	0
MGT018B	SUBTASKS	ADA	1912	288	758	112640	9	0	0	158	32	79	7	29	3
MGT018S	SUBTASKS_	ADA	194	142	26	1536	0	0	0	1	0	0	0	7	0
MGT019P	MTL_EDITOR	ADA	27	19	4	1024	1	0	0	0	0	0	0	2	0
MGT023B	PTR_SUPPORT	ADA	894	151	414	90624	0	0	0	69	45	19	14	15	11
MGT023S	PTR_SUPPORT_	ADA	337	232	52	5632	0	0	0	5	1	0	0	6	0
MGT024B	OTHER_TRS	ADA	2748	550	904	136192	3	0	0	388	44	64	17	18	9
MGT024S	OTHER_TRS_	ADA	264	177	42	3584	0	0	0	12	3	0	0	6	0
MGT025P	TPP_EDITOR	ADA	29	20	4	1024	1	0	0	0	0	0	0	2	0
MGT028P	OTR_EDITOR	ADA	31	19	5	1536	1	0	0	2	0	1	0	2	0

CMS NUMBER	UNIT NAME	LANGUAGE	TOTAL LINES	COMMENT LINES	SLOC	OBJECT CODE FILE SIZE (BYTES)	MIX ELEMENTS MOPR	MINT	MREA	MONL	MDAT	MSTR	MMAT	NUMBER OF UNITS CALLED	NUMBER OF UNITS CALLED BY
MGT030B	ITR	ADA	3659	835	1358	232448	2	0	0	414	224	37	105	37	50
MGT030S	ITR_	ADA	170	135	16	2560	0	0	0	2	5	0	0	4	47
MGT031B	EVENTS_	ADA	154	118	24	2560	0	0	0	2	2	2	0	5	0
MGT031S	EVENTS	ADA	1588	84	703,	140800	0	0	0	183	109	49	31	28	25
MGT032B	ATR	ADA	1975	272	900	165376	7	0	0	189	244	39	27	29	17
MGT032S	ATR_	ADA	224	131	69	10240	0	0	0	7	7	16	10	10	0
MGT033B	TRAINING_HIST	ADA	1579	86	733	140800	0	0	0	224	93	53	36	25	13
MGT033S	TRAINING_HIST_	ADA	369	228	75	6656	0	0	0	7	19	1	3	8	0
MGT038B	SCHEDULER	ADA	591	139	214	19968	1	0	0	27	18	12	13	12	4
MGT038S	SCHEDULER_	ADA	160	122	19	2048	0	0	0	0	2	0	0	3	0
MGT039B	QUAL_ASS	ADA	1770	608	623	152576	2	0	0	51	97	4	12	26	1
MGT039S	QUAL_ASS_	ADA	56	50	3	1024	0	0	0	0	0	0	0	1	0
MGT040B	EVENT_TYPES	ADA	666	156	240	38400	0	0	0	64	48	0	43	14	25
MGT040S	EVENT_TYPES_	ADA	401	241	104	11776	0	0	0	7	12	3	6	10	0
MGT042B	GENERAL_TRNG	ADA	868	124	346	52736	0	0	0	129	77	34	43	14	8
MGT042S	GENERAL_TRNG_	ADA	207	128	45	6144	0	0	0	7	8	2	14	6	0
MGT043B	FILE_OPS	ADA	248	72	76	8192	1	0	0	16	13	2	0	7	21
MGT043S	FILE_OPS_	ADA	133	91	23	1536	0	0	0	2	5	0	0	2	0
MGT044B	PME_EDIT	ADA	754	104	290	38912	0	0	0	29	11	6	18	0	0
MGT044S	PME_EDIT_	ADA	88	49	13	1536	0	0	0	0	0	0	0	0	0
MGT045B	FORMAL_TRNG	ADA	1051	148	432	90624	0	0	0	119	46	38	25	13	9
MGT045S	FORMAL_TRNG_	ADA	255	161	53	4608	0	0	0	6	9	5	3	7	0
MGT046B	ECI_CDC	ADA	640 $	b1; %)	241	73216	0	0	0	44	37	10	17	12	9
MGT046S	ECI_CDC_	ADA	240	155	43	4096	0	0	0	6	9	2	4	7	0
MGT047B	ANCADDCON_	ADA	252	161	52	4608	0	0	0	7	9	6	3	7	0
MGT047S	ANCADDCON	ADA	817	148	339	83456	0	0	0	61	40	26	16	14	8
MGT048B	QC_EVAL	ADA	1283	445	420	63488	2	0	0	6	43	15	22	19	2
MGT048S	QC_EVAL_	ADA	189	103	45	9216	0	0	0	0	6	0	4	8	0
MGT049B	MQC_EVAL	ADA	749	151	266	35840	0	0	0	92	39	9	11	15	1
MGT049S	MQC_EVAL_	ADA	83	74	3	1024	0	0	0	0	0	0	0	0	0
MGT050B	OTR_EDIT	ADA	1288	252	390	144896	0	0	0	147	20	47	4	11	8
MGT050S	OTR_EDIT_	ADA	292	181	47	7168	0	0	0	13	1	2	0	5	0
MGT051B	OTR_PRINT	ADA	303	32	107	14336	0	0	0	77	1	5	2	8	2

Table A-8. AOTS File Metric List. (Continued)

CMS NUMBER	UNIT NAME	LANGUAGE	TOTAL LINES	COMMENT LINES	SLOC	OBJECT CODE FILE SIZE (BYTES)	MIX ELEMENTS MOPR	MINT	MREA	MONL	MDAT	MSTR	MMAT	NUMBER OF UNITS CALLED	NUMBER OF UNITS CALLED BY
MGT051S	OTR_PRINT_	ADA	196	119	27	3584	0	0	0	16	1	0	0	5	0
MGT052B	MEMBER	ADA	101	38	33	4608	0	0	0	0	0	3	0	5	1
MGT052S	MEMBER_	ADA	79	59	8	1536	0	0	0	0	0	0	0	1	0
MGT054B	ITR_EVENTS	ADA	4730	780	1926	388096	0	0	0	252	140	63	80	34	9
MGT054S	ITR_EVENTS_	ADA	273	196	44	3072	0	0	0	4	9	0	2	7	0
MGT055B	MAP_TASK	ADA	3296	574	1426	102912	0	0	0	214	60	45	27	25	5
MGT055S	MAP_TASK_	ADA	547	409	83	3072	0	0	0	4	9	0	0	10	0
MGT057P	FINAL_MTL_GEN	ADA	1576	419	412	54784	3	0	0	51	22	9	1	25	0
MGT058S	MTL_BUFFER_	ADA	60	29	18	2560	0	0	0	0	1	0	0	4	0
MGT059S	TASK_LIST_DEF_	ADA	135	55	33	4096	0	0	0	0	0	0	0	4	0
MGT060B	FINAL_MTL_FUNCTIONS	ADA	4117	1051	1152	139776	1	0	0	167	91	44	38	30	3
MGT060S	FINAL_MTL_FUNCTIONS_	ADA	586	364	89	13312	0	0	0	3	6	0	0	11	0
MGT061P	MTL_FUNCTION_SELECT	ADA	659	193	158	18944	0	0	0	26	8	7	0	13	1
MGT062B	ITR_PRINT	ADA	611	177	258	43520	0	0	0	22	16	8	6	15	6
MGT062S	ITR_PRINT_	ADA	239	150	54	6144	0	0	0	2	10	0	0	5	0
MGT063P	ITR_BG	ADA	1471	450	512	110080	3	0	0	87	45	15	7	25	0
MGT064B	MAP_DISPLAY	ADA	643	159	219	22528	3	0	0	65	36	0	10	20	1
MGT064S	MAP_DISPLAY_	ADA	224	152	43	4608	0	0	0	22	3	0	0	6	0
MGT065B	ITR_TYPES	ADA	938	125	450	38400	0	0	0	44	134	23	8	16	30
MGT065S	ITR_TYPES_	ADA	858	385	260	31232	0	0	0	9	31	1	36	15	0
MGT066P	CSTEDIT	ADA	81	21	27	3584	0	0	0	5	0	0	0	6	1
MGT067P	EXTRACT	ADA	894	526	223	27136	0	0	0	23	50	17	1	16	0
MGT068C	INSTRUCTION MANUAL	VAX DCL	214	N/A	214	N/A	74	0	0	0	35	0	0	N/A	N/A
MGT085P	DAILY_RUN	ADA	229	94	63	7168	0	0	0	0	10	0	3	10	0
MGT085P	QCRUN	ADA	212	90	57	6656	1	0	0	0	8	2	8	N/A	N/A
MGT086B	GPTR_EDIT	ADA	617	186	198	27648	0	0	0	59	12	5	1	13	3
MGT086S	GPTR_EDIT_	ADA	54	41	4	1536	0	0	0	0	0	0	0	2	0
MGT087B	OPTR_EDIT	ADA	1221	197	519	153088	0	0	0	115	55	9	13	23	6
MGT087S	OPTR_EDIT_	ADA	211	145	30	4608	0	0	0	3	2	0	0	7	0
MGT088B	ITR_ECI	ADA	1148	267	316	92160	0	0	0	133	45	17	69	10	1
MGT088S	ITR_ECI_	ADA	87	66	9	2048	0	0	0	0	2	0	0	2	0
MGT089S	TRAINING_TYPES_	ADA	101	26	35	13312	0	0	0	0	1	18	6	6	0
MGT090P	BURSTER	ADA	166	68	51	7168	0	0	0	2	20	0	1	2	0

Table A-8. AOTS File Metric List. (Continued)

CMS NUMBER	UNIT NAME	LANGUAGE	TOTAL LINES	COMMENT LINES	SLOC	OBJECT CODE FILE SIZE (BYTES)	MIX ELEMENTS MOPR	MINT	MREA	MONL	MDAT	MSTR	MMAT	NUMBER OF UNITS CALLED	NUMBER OF UNITS CALLED BY
MGT091S	MTL_BASIC_	ADA	61	21	20	4608	0	0	0	0	0	0	0	3	0
MGT092B	MTL_COMMON	ADA	1345	177	533	110080	4	0	0	61	23	15	30	26	3
MGT092S	MTL_COMMON_	ADA	195	172	15	1536	0	0	0	0	1	0	0	2	0
MGT093B	MTL_INFREQUENT	ADA	1788	143	866	84480	10	0	0	153	50	78	20	31	6
MGT093S	MTL_INFREQUENT_	ADA	243	172	49	7168	0	0	0	3	0	1	0	8	0
MGT094B	MTL_PRINT	ADA	1345	126	576	60928	0	0	0	574	36	27	4	23	2
MGT094S	MTL_PRINT_	ADA	64	47	9	1536	0	0	0	4	0	0	0	1	0
MGT095B	MTL_TYPES	ADA	595	56	251	21504	0	0	0	40	15	4	1	10	28
MGT095S	MTL_TYPES_	ADA	403	187	140	17920	4	0	0	5	3	0	3	14	0
MGT096B	SUB_FUNC	ADA	256	61	120	9216	0	0	0	3	9	14	3	18	3
MGT096S	SUB_FUNC_	ADA	161	109	27	1536	0	0	0	0	0	0	0	7	0
MGT097B	SUB_ITEMS	ADA	187	50	87	10240	0	0	0	15	8	4	1	13	1
MGT097S	SUB_ITEMS_	ADA	75	53	8	1536	0	0	0	0	0	0	0	4	0
MGT098B	SUB_PRINT	ADA	374	80	135	19968	0	0	0	139	2	28	5	19	2
MGT098S	SUB_PRINT_	ADA	73	49	11	1536	0	0	0	7	0	0	0	5	0
MGT099S	SUB_TYPES_	ADA	108	44	33	6656	0	0	0	0	0	2	0	8	0
MGT100B	SUB_UTIL	ADA	963	103	465	44544	1	0	0	104	47	34	17	16	8
MGT100S	SUB_UTIL_	ADA	356	215	66	6656	0	0	0	6	5	4	0	6	0
MGT101P	UPDATE_TRAINING_RECORDS	ADA	1160	342	428	35328	0	0	0	45	49	8	19	17	0
MGT102B	DEBUG_PKG	ADA	1210	270	365	50688	0	0	0	46	2	4	14	7	4
MGT102S	DEBUG_PKG_	ADA	277	177	34	4608	0	0	0	0	0	0	0	1	0
MGT103B	SINGLE_LINKED_LIST_PKG	ADA	90	25	29	1024	0	0	0	0	0	0	0	0	1
MGT103S	SINGLE_LINKED_LIST_PKG_	ADA	102	49	23	N/A	0	0	0	0	0	0	0	0	0
MGT104C	IMPLEMENT MTL(104)	VAX DCL	16	N/A	16	N/A	2	0	0	0	12	0	0	N/A	N/A
MGT105C	IMPLEMENT MTL(105)	VAX DCL	6	N/A	6	N/A	2	0	0	0	4	0	0	N/A	N/A
MGT106C	IMPLEMENT MTL(106)	VAX DCL	22	N/A	22	N/A	10	0	0	0	12	0	0	N/A	N/A
MGT107C	IMPLEMENT MTL(107)	VAX DCL	22	N/A	22	N/A	10	0	0	0	12	0	0	N/A	N/A
MGT108C	IMPLEMENT MTL(108)	VAX DCL	22	N/A	22	N/A	10	0	0	0	12	0	0	N/A	N/A
MGT109C	IMPLEMENT MTL(109)	VAX DCL	22	N/A	22	N/A	10	0	0	0	12	0	0	N/A	N/A
MGT110C	IMPLEMENT MTL(110)	VAX DCL	17	N/A	17	N/A	8	0	0	0	9	0	0	N/A	N/A
MGT111C	IMPLEMENT MTL(111)	VAX DCL	172	N/A	172	N/A	37	0	0	0	113	0	0	N/A	N/A
MGT112B	ITR_RECORD_UPDATE	ADA	1050	254	395	33792	0	0	0	20	62	6	44	17	1
MGT112S	ITR_RECORD_UPDATE_	ADA	74	57	5	1024	0	0	0	0	0	0	0	1	0

Table A-8. AOTS File Metric List. (Continued)

CMS NUMBER	UNIT NAME	LANGUAGE	TOTAL LINES	COMMENT LINES	SLOC	OBJECT CODE FILE SIZE (BYTES)	MIX ELEMENTS MOPR	MINT	MREA	MONL	MDAT	MSTR	MMAT	NUMBER OF UNITS CALLED	NUMBER OF UNITS CALLED BY
MGT113B	REPORT_TYPES	ADA	181	33	70	10240	0	0	0	35	5	6	0	8	3
MGT113S	REPORT_TYPES_	ADA	121	19	33	11264	0	0	0	0	2	6	4	5	0
MGT114P	REPORTS	ADA	1559	182	632	110592	1	0	0	201	43	60	22	20	1
MGT115P	PRINT_REPORTS	ADA	230	67	79	4096	0	0	0	27	5	4	7	9	0
MGT116B	VALIDATE_PKG	ADA	2156	631	575	55808	0	0	0	129	8	15	13	19	1
MGT116S	VALIDATE_PKG_	ADA	75	51	8	2048	0	0	0	4	0	0	0	2	0
MGT117C	IMPLEMENT MTL(117)	VAX DCL	7	N/A	7	N/A	3	0	0	0	2	0	0	N/A	N/A
MGT118P	POS_TO_AOTS	ADA	567	72	296	42496	1	0	0	62	47	45	12	18	0
MGT120P	CHECK_CHANGES	ADA	828	59	179	54784	0	0	0	232	8	23	2	13	2
MGT???X	SAS PROGRAMS	SAS	7540	N/A	3053	N/A	154	0	0	0	0	44	14	N/A	N/A
SUP001B	SCREEN_IO	ADA	4774	356	2748	201728	0	0	0	332	230	48	100	15	88
SUP001S	SCREEN_IO_	ADA	1468	1017	268	11264	0	0	0	10	43	1	3	4	0
SUP004B	LISTPKG	ADA	289	106	108	1024	0	0	0	11	2	0	2	1	1
SUP004S	LISTPKG_	ADA	219	148	31	N/A	0	0	0	2	1	0	0	1	0
SUP005B	EDFIELD_	ADA	308	220	67	7168	0	0	0	7	2	0	2	3	0
SUP005S	EDFIELD	ADA	2056	595	861	68608	2	0	0	100	26	8	44	5	10
SUP006B	UTILITIES	ADA	297	104	97	8704	1	0	0	0	4	8	14	4	15
SUP006S	UTILITIES_	ADA	182	126	26	1536	0	0	0	0	1	1	0	2	0
SUP007B	DATE_TIME	ADA	812	205	394	36352	1	0	0	25	25	25	53	6	71
SUP007S	DATE_TIME_	ADA	346	215	84	6656	0	0	0	1	1	0	1	4	0
SUP009B	HELPTEXT	ADA	1007	102	474	58880	0	0	0	117	63	29	7	9	3
SUP009S	HELPTEXT_	ADA	60	44	7	1024	0	0	0	0	1	0	0	1	0
SUP013B	KEYBOARD	ADA	218	38	59	7680	2	0	0	4	0	0	0	7	6
SUP013S	KEYBOARD_	ADA	86	56	8	1536	0	0	0	1	0	0	0	0	0
SUP014B	EXPRESSION_HANDLER	ADA	1554	299	692	1024	0	0	0	75	29	8	37	8	5
SUP014S	EXPRESSION_HANDLER_	ADA	229	129	52	N/A	0	0	0	1	17	2	2	3	0
SUP018A	COM_CODE	ASM	389	89	238	N/A	0	0	0	0	0	0	0	N/A	N/A
SUP019A	GRAF_CODE	ASM	1028	334	659	N/A	0	0	0	0	0	0	0	N/A	N/A
SUP024B	RANDOM_PASSWORD	ADA	209	78	78	9728	0	0	0	14	9	2	3	8	4
SUP024S	RANDOM_PASSWORD_	ADA	74	59	11	3072	0	0	0	1	0	0	0	2	0
SUP026B	USER_TYPE	ADA	1724	358	715	184832	19	0	0	139	81	32	28	17	67
SUP026S	USER_TYPE_	ADA	659	521	100	11776	2	0	0	11	4	15	0	8	0
SUP027P	AOTS_LOGON	ADA	1612	435	683	105984	96	0	0	159	88	42	7	46	0

Table A-8. AOTS File Metric List. (Continued)

CMS NUMBER	UNIT NAME	LANGUAGE	TOTAL LINES	COMMENT LINES	SLOC	OBJECT CODE FILE SIZE (BYTES)	MIX ELEMENTS MOPR	MINT	MREA	MONL	MDAT	MSTR	MMAT	NUMBER OF UNITS CALLED	NUMBER OF UNITS CALLED BY
SUP028B	ITEM_EDITOR	ADA	1361	256	605	1024	0	0	0	136	20	2	29	5	20
SUP028S	ITEM_EDITOR_	ADA	291	204	69	N/A	0	0	0	8	0	0	0	4	0
SUP029B	ITEM_SUP_TYPES	ADA	279	62	131	10240	0	0	0	14	1	2	15	3	22
SUP029S	ITEM_SUP_TYPES_	ADA	285	194	74	7168	0	0	0	5	1	0	1	3	0
SUP030S	GLOBALS_	ADA	94	10	75	4096	0	0	0	0	0	0	0	1	0
SUP031B	PRINT_SUPPORT	ADA	708	165	337	38912	0	0	0	58	19	27	9	15	68
SUP031S	PRINT_SUPPORT_	ADA	369	281	69	5632	0	0	0	15	3	0	0	3	0
SUP032B	LOG_IO	ADA	268	102	73	11264	2	0	0	10	7	6	0	6	5
SUP032S	LOG_IO_	ADA	61	45	7	1024	0	0	0	0	0	0	0	0	0
SUP033B	DMSF	ADA	1586	525	508	41472	0	0	0	2	4	5	16	12	39
SUP034P	CALL_DCL	ADA	27	7	11	3072	1	0	0	1	0	1	0	4	0
	A01USRNAM	ADA	25	2	12	2560	2	0	0	2	1	1	0	3	4
	A01USRNAM_	ADA	110	39	51	9216	0	0	0	1	0	0	0	5	0
	CA1CRA	ADA	643	82	307	39424	2	0	0	107	2	13	11	12	1
	CA1CRA_	ADA	95	76	7	1536	0	0	0	0	0	0	0	2	0
	CABGEN	ADA	641	124	301	27648	30	8	0	36	0	10	15	7	5
	CABGEN_	ADA	276	179	54	12288	0	0	0	0	0	0	0	4	0
	CACR	ADA	829	167	318	28672	2	0	0	4	0	6	29	7	7
	CACR_	ADA	126	93	21	2048	0	0	0	0	0	0	0	3	0
	CAGEN	ADA	768	69	406	42496	0	1	0	78	0	21	22	8	15
	CAGEN_	ADA	540	417	67	10240	0	0	0	0	0	0	0	2	0
	CTCR_	ADA	197	69	84	13824	0	0	0	0	0	0	0	2	0
	CTEXP_	ADA	204	92	69	12288	0	0	0	0	0	0	0	4	0
	GRLIB	ADA	1886	341	915	57344	5	0	0	69	14	1	389	8	2
	GRLIB_	ADA	130	110	9	1536	0	0	0	0	0	0	0	2	0
	GRTYPES	ADA	15	0	7	1536	0	0	0	0	0	0	1	2	6
	GRTYPES_	ADA	264	79	135	13824	0	0	0	0	0	0	2	3	0
	LGLIB	ADA	83	3	55	6656	0	0	0	8	0	0	0	7	3
	LGLIB_	ADA	79	60	6	1536	0	0	0	0	0	0	0	7	0
	LGTYP_	ADA	217	50	97	14336	5	0	0	0	0	0	0	9	0
	SD0KEYTYP_	ADA	22	4	11	1536	0	0	0	0	0	0	0	1	0
	SD3SR_	ADA	21	3	11	3584	0	0	0	0	0	0	0	3	0
	SD4PASS_	ADA	79	28	25	4096	0	0	0	0	0	0	0	1	0

Table A-8. AOTS File Metric List. (Continued)

CMS NUMBER	UNIT NAME	LANGUAGE	TOTAL LINES	COMMENT LINES	SLOC	OBJECT CODE FILE SIZE (BYTES)	MIX ELEMENTS MOPR	MINT	MREA	MONL	MDAT	MSTR	MMAT	NUMBER OF UNITS CALLED	NUMBER OF UNITS CALLED BY
	SDINTSTO_	ADA	18	3	8	1024	0	0	0	0	0	0	1	1	0
	SWAP_EMULATORS	ADA	792	242	235	22016	5	0	0	31	9	16	10	12	1
	SWAP_EMULATORS_	ADA	140	92	8	1536	0	0	0	1	0	1	0	2	0
	TSOTYPES_	ADA	399	99	209	22528	0	0	0	0	0	0	0	2	0
	SACCESS	ADA	37	0	23	2560	0	0	0	0	0	0	0	3	5
	SACCESS	ADA	104	42	40	4096	0	0	0	0	0	0	0	1	0
	TOTALS		166,027	41,624	62,433	7,774,208	790	34	0	11212	4428	2592	2778	2086	1322

The number of comment lines does not include side comments in the code. SLOC is defined here as the number of executable statements in the referenced unit. For units programmed in Ada these are statements that end in a semi-colon (;). The object code file size is in bytes.

The mix elements were counted in support of a cost analysis of a future FSD AOTS system. The mix elements are defined in Table A-9.

The last two columns in Table A-8 are presented to show how much interaction occurs between the units. The number of units called is a count of the units listed in the context (WITH, USE) statements of each unit. It is assumed that if a unit is referenced in the context statement it is called somewhere in the body of the program. The last column is the number of times a unit's name is found in other units context statements, i.e. the number of times it is called by other programs.

It must be recognized that the numbers presented here are approximate since development of the AOTS was occurring both during and after the counting process took place. Several units did not exist at the time counting took place and therefore Table A-8 does not list them. The symbol N/A in Table A-8 denotes where a data item was not counted for the referenced unit.

The classifying and counting was accomplished using interactive programs that were developed for that purpose. The programs searched the AOTS Ada source code files for specific characters or strings of characters. In the case of counting the mix elements, if the program could not classify a line then the user was presented the line and asked to classify the line.

C. Software Development Environment. The prototype AOTS was developed on a VAX 8600 series computer system under the VAX/VMS operating system. The AOTS is written primarily in the VAX Ada programming language. Several software development tools, utilities, and application programs resident on the VAX were used in the development of the AOTS. These include VAX ACS, VAX CMS, VAX VMS, KERMIT, VAX FORTRAN. The following paragraphs describe in more detail the various development utilities, personnel, facilities, and general development procedures for the prototype AOTS.

1. VAX/VMS. The VAX/Virtual Memory System (VMS) operating system supports all VAX series computers in both time-sharing and production environments. The system permits an absolute limit of 8192 concurrent processes. User access to VMS is by the DIGITAL Command Language (DCL). On-line documentation is

obtained utilizing the Help utility. VMS provides a number of line-mode editors, a line-mode/character-mode editor (EDT), and a character-mode editor (TPU). EDT logs changes and replays them on demand should a failure occur during an editing session. VMS provides a comprehensive set of tools for developing programs including editors, a file difference utility, programming languages, a linker, a librarian, a symbolic debugger, and a patch utility. The assembly level VAX MACRO language is also available.

Table A-9. Mix Elements.

Source: Price S Users Manual, Table 118, RCA Corp, 1977

OPERATING SYSTEMS (MOPR): Task management. Memory management. Heavy Hardware Interface. Many interactions. High reliability and strict timing requirements.

INTERACTIVE OPERATIONS (MENT): Real time man/machine interfaces. Human engineering construction and error protection very important.

REAL TIME COMMAND AND CONTROL (MREA): Machine to machine communications under tight timing constraints. Queuing not practicable. Heavy hardware interface. Strict protocol requirements.

ON-LINE COMMUNICATIONS (MONL): Machine to machine communications with queuing allowed. Timing restrictions not as restrictive as with real time command and control.

DATA STORAGE AND RETRIEVAL (MDAT): Operation of data storage devices. Data base management. Secondary storage handling. Data blocking and deblocking. Hashing techniques. Hardware oriented.

STRING MANIPULATION (MSTR): Routine applications with no overriding constraints. Not oriented toward mathematics. Typified by language compilers, sorting, formatting, buffer manipulation, etc.

MATHEMATICAL OPERATIONS (MMLAT): Routine mathematical applications with no overriding constraints.

2. VAX Ada. VAX Ada (a registered trademark) is Digital's government-validated implementation of the full ANSI-MIL-STD-1815-A Ada language. VAX Ada is integrated in the VAX/VMS common language environment. Full symbolic debugging and VAX/VMS system services and utilities are available to programs written in Ada. VAX Ada programs can invoke modules written in other VAX/VMS languages. Additionally, programs written in other languages can invoke VAX Ada modules. The Ada language implements the concept of program libraries and each Ada compilation unit must be associated with a library of related Ada modules. When a user creates an Ada library, the VAX Ada includes the standard system units into the new library. Once the user compiles a newly developed unit, that unit's compilation products are also entered into the library.

3. VAX CMS. The DEC Code Management System (CMS) is a program librarian for software development and evolution. It is comprised of a set of commands enabling software developers to manage the files of an ongoing project. CMS stores source code

files in a configuration management library, keeps track of changes made to the files, and records user access to the files. CMS files include not only Ada source files, but any file on which the configuration manager maintains configuration status accounting.

4. VAX ACS. ACS is the VAX Ada program library management utility. It performs Ada code library management functions, and provides an interface to the Ada compiler and linker utilities.

5. VAX FORTRAN. VAX FORTRAN is an implementation of full-language FORTRAN-77, conforming to ANSI-X3.9-1978 American National Standard FORTRAN. The shareable, re-entrant compiler operates under the VAX/VMS operating system. It takes advantage of the VAX floating-point and character string instruction set and the VAX/VMS virtual memory system.

6. ZDS ZSTEMpc. The ZSTEMpc version 2 application program is a powerful, high performance emulator that enabled the use of the Zenith Z-248 personal computers as DEC VT100 terminals. Almost all VT100 terminal functions and many VT102 extensions are supported. ZSTEM also supports microcomputer to microcomputer and microcomputer to mainframe communications using the ASCII, Xmodem, and Kermit protocols.

Kermit is a nearly error-free file transfer protocol developed at Columbia University that was initially used to communicate between microcomputers and mainframes. The Kermit protocol allows the interchange of any 7 bit or 8 bit data file, including object code and executable programs. Communications take place over ordinary, serial terminal connections. The communications are asynchronous and can be either hard-wired or through a modem. The packet length is variable up to a maximum of 96 characters. Packet flow is bi-directional. All transmissions are in displayable ASCII characters.

7. Personnel and Facilities. The AOTS software support resources are illustrated in Figure A-9. These resources represent the personnel, systems, and facilities used to provide support since the 1 August 1988 version 2.0 AOTS Baseline Initial Operational Capability. The hardware and facilities are discussed in more detail in section II A of this appendix. Design, coding, and support of the AOTS software was performed by the Douglas Aircraft Company (DAC). The DAC team consisted of three support sub-groups whose managers reported to the DAC AOTS Program Manager. At the time of this report, DAC had four software engineers and a chief software engineer. In addition, DAC had five Instructional Systems Developers. One of the five software engineers and one of the ISD team members was located in Building 1808. One software engineer

was also physically located in the AFHRL at Brooks AFB, Texas. Each engineer had two Z-248s and access to a local laser printer. Each group also had access to another laser printer connected to the VAX 8650.

8. Development Procedures. The AOTS prototype developer established several separate versions of the AOTS software that are resident in different directories. Roughly, these versions were stored in directories called development, test, and production. Individual software engineers developed and modified files in the development directory. When the developed unit was individually tested by the engineer and was ready to pass on to the system tester, it was entered into the test directory. The testing staff periodically performed system testing in the test directory. Finally, when a successful system test is run, the executable image is copied into the production directory.

Associated with the development directory were subdirectories for source, CMS, and Ada library files. The development directories contained the latest, albeit untested, versions of the units. When the software engineer received an authorization to modify a unit, the engineer checked the unit out of the CMS directory and modified it in the engineer's source directory. When satisfied with the change, the modified unit was returned to the CMS directory and the compilation products updated the Ada library in the development library. The changes could be tested by individuals using an executable image stored in this area.

The test directory also contained sub-directories for CMS and Ada libraries. Developed and modified units of Ada source code were entered into the respective test sub-directories when they were deemed ready for system testing. Periodically a system test is copied into the production directory. The production environment consists of the executable image and the production data base files. There was no need for Ada or CMS libraries in the production directories.

9. Software Problem Reports (SPR). A system for reporting and tracking software problems was established during the development of AOTS. AOTS problems were reported on SPR forms (SPR AF Form 1775) which were logged and given a unique identification number. The SPRs were analyzed by software engineers to determine the complexity of the problem reported, the solution, and the risk associated with the implementation of the solution. The evaluated SPRs were then presented to a Configuration Management Board, which met weekly, for disposition. Status of SPRs was tracked using the Lotus 123 application software package.

Lotus 123 is an integrated spreadsheet applications program. Lotus 123 can be hosted on mainframes or microcomputers operating under the MS-DOS operating system. Lotus 123 supports automatic computation and updating of tabularized data and provides graphics capabilities.

10. Operational AOTS VAX Environments. The VAX environments that contained the data required to allow users to fully apply AOTS capabilities were:

(a) AT01 (1ST Development). This area was further broken down into three components, Active Duty, Reserve, and the Air National Guard.

(b) AT02 (Active Duty). This environment contained the actual data that was used by the AOTS work centers for the active duty participants.

(c) AT03 (Reserve). This environment contained the actual data that was used by the AOTS work centers for the Air Force Reserve participants.

(d) AT04 (Air National Guard). This environment contained the actual data that was used by the AOTS work centers for the Air National Guard participants.

(e) AT05 (1ST Test). This environment contained the data for use by the Instructional Systems Team (IST) to train users on the AOTS.

APPENDIX B FULL SCALE DEVELOPMENT SYSTEM

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SECTION I - FULL SCALE DEVELOPMENT SYSTEM REQUIREMENTS

A. Overview. The objective of the FSD AOTS will be to produce a computer program that can be adopted and applied by any Air Force organization. The FSD program is expected to deliver a software "shell" to that user. The FSD program may not provide MTLs, GPTRs, OPTRs, training material, or other data that is required in a specific implementation of the AOTS system.

The FSD program user will require a way to acquire or develop courseware and provide system maintenance functions. Recognizing that any engineered system requires maintenance and update during its useful life, the AOTS will require a support organization to evaluate necessary software and hardware design changes that inevitably occur. The maintenance organization would have engineering responsibility to include issuing standards, operating instructions, configuration management, and other system engineering areas.

The purpose of this analysis is to hypothesize reasonable approaches and architectures for implementing an FSD AOTS and to evaluate the effort, and thereby the cost, of adopting each of these approaches. Three different approaches are discussed in this Appendix. They are 1) using the prototype software and the existing centralized hardware architecture by making only minimal changes to produce an FSD version, 2) porting the prototype software to a standalone personal computer hardware environment, and making the necessary functional enhancements to produce an FSD AOTS, and 3) converting the prototype software to a networked environment and making necessary functional enhancements.

In addition to the three scenarios above, this Appendix discusses the possibility of improving the supportability of the prototype AOTS by modifying the tools and procedures used to manage the AOTS software.

The user is expected to interact with the FSD AOTS at a terminal, where the user can take Computer Aided Instruction (CAI), develop training material, or perform management or evaluation functions. This user perspective is displayed in Figure B-1. The user has available the functions listed inside the outer rectangle of Figure B-1. The inner boxes of the figure contain interfaces that are not visible to the user. These internal interfaces are available to the maintainer and aid the software system supportability.

This Appendix in organized into two sections. Section I begins with a discussion of aspects common to the three scenarios. The section continues on to discuss each scenario. Section I ends

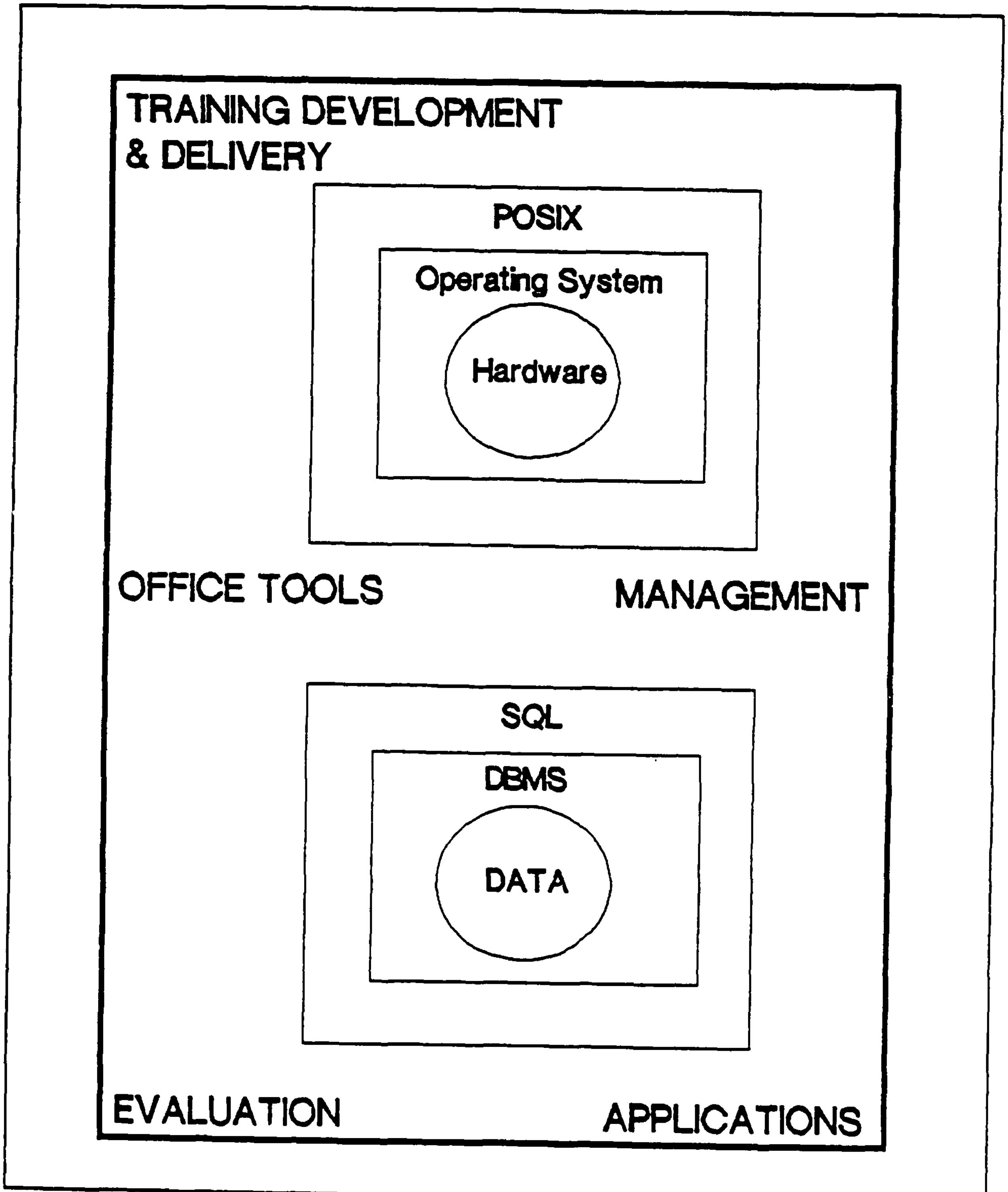

Figure B-1. Software Architecture, Users View with a discussion of suggested supportability enhancements to the prototype system.

Section II provides analyses on each of the three scenarios, and draws some estimates and ramifications on the amount of effort required to evolve the prototype source code into the envisioned future system. Each subsection of Section II is designed to stand on its own, and material from Section I is repeated where appropriate to avoid a cover-to-cover reading in order to follow the methodology.

B. Prototype Software Modification. A cost-effective production system option may be the use of the prototype software on the current centralized host computer hardware with a minimum set of code changes to enter the field. These minimum changes required to modify the prototype to accommodate the above scenario include modifying the software to support 250-400 AFSs, provide interfaces to other Air Force Systems, and other enhancements. Other enhancements include adding system backup and restore capability, interfaces, and re-writing all non-Ada code in this scenario. Added interfaces which are discussed include Defense Data Network (DDN), Structured Query Language (SQL), and Portable Operating System Interface for Computer Environments (POSIX), or another appropriate operating system interface.

In its present form, the POSIX standard focuses primarily on the C Language interface to the operating system, but this standard is the first of a group of proposed standards known collectively as POSIX. The Ada Language Bindings to POSIX are currently in development by an IEEE Computer Society Working Group.

The Air Force Military Personnel Center (AFMPC) is expected to periodically provide AOTS with personnel and education data to update the system. This update may be via a magnetic tape. Personnel data must be updated quite often or the user will soon become disenchanted with old data. A well-managed system would be updated possibly three times a week or even in real time. To avoid AFMPC manually handling a myriad of requests from systems around the world, the data would not come directly from AFMPC, but from the servicing Consolidated Base Personnel Office (CBPO). The AOTS system administrator may receive the tape from the CBPO, and then run the PDS_TO_AOTS program to install the new information in the ATR data base files.

Communication with the Air Force Occupational Measurement Center (USAFOMC) and Core Automated Maintenance System (CAMS) data might be a manual task. Again the file may be obtained in electronic format via tape, disk, or modem, but the system administrator may be required to manually install the data.

USAFOMC may be a logical source for Master Task Lists (MTLs). If this is so, the system administrator may receive a new MTL for an Air Force Specialty (AFS) perhaps semiannually and install it.

Multiple terminals are available to users. The users can manage training, take training, or use office tools. An ad hoc report generator might be available to support the user.

When a trainee enters the system, the system administrator would manually enter him into the system and insure that the new trainee is included in future personnel data updates. Upon leaving the organization, the trainee is provided with both a paper and magnetic copy and of his/her ATR information.

The functional modifications required to convert the prototype AOTS in to an FSD version under the existing centralized architecture are discussed below.

1. Generalize to Work for Other AFSs. A basic requirement for the reuse or porting of the Prototype AOTS is the expansion of the prototype present capability to manage a maximum of four Air Force Specialty Codes to a much larger number. The present prototype installed on the VAX 8650 should accommodate 250-400 Air Force Specialty codes.

It is considered unlikely that the user of AOTS will be satisfied with exactly four AFSs. Most Air Force organizations require the skills of many AFSs and the list of current AFSs for the Air Force contains hundreds of possibilities. Further, the Air Force continually revises the list of AFSs as the mission and technologies change within the force structure.

Probably the maximum number of AFSs will not be implemented on any one AOTS system, but some reasonable number of them would. That list should be easily modified within AOTS as the AFSs the user decides to support change with time. Currently the prototype creates and manipulates 13 AFS-unique data files for each AFS. Each user of the prototype is identified by one of the four supported AFSs. The software uses that identified AFS to process AFS-unique actions for that user. Within the AOTS program file structure, a numbering convention can be devised for AFS-related files based on the unique AFS number and directories can be established to hold them.

2. Interface to Other Air Force Systems. One probable change from the prototype to the FSD AOTS is the implementation of interfaces to other Air Force systems. The paragraphs below identify the changes required to provide the AOTS software with interfaces to relevant Air Force systems.

(a) Personnel Concept III. Personnel Concept III (PC III) is a large computer-based system that decentralizes the processing of personnel data to the unit level through the use of remote terminal work centers in unit orderly rooms connected to a base- level mainframe. The unit-level work centers are staffed by personnel specialists and perform approximately 64 personnel functions that used to be performed at the Consolidated Base Personnel Office (CBPO).

The Automated Personnel Data System II (APDS II) is hosted on a base-level Phase IV Sperry (UNISYS) 1100/60 mainframe. PC III will use the AT&T 3B2 computer with the main processor at Randolph AFB Texas, and file servers at bases. A communications network will connect the base computer to other Air Force facilities.

(b) USAF Occupational Measurement Center. The Air Force Occupational Measurement Center (OMC) plans and analyzes Air Force training. The OMC conducts a comprehensive occupational analysis program that includes a listing of all tasks required to be performed in a AFS.

(c) Core Automated Maintenance System. The Core Automated Maintenance System (CAMS) is a large, real-time, on-line system used at the unit level to manage maintenance equipment and personnel resources. CAMS also provides much of the maintenance data needed by major commands, Air Force Logistics Command, Headquarters USAF, and other agencies to manage and track maintenance resources worldwide. The system applies to aircraft, missile, and Communication-Electronics (C-E) maintenance.

CAMS provides the capability for maintenance personnel to communicate to a central base-level computer via remote terminals in selected maintenance work areas. CAMS is designed to operate on the standard base-level Phase IV x2/x3 configuration using Sperry (UNISYS) 1100/60 Computer Systems. The remote terminals are Sperry Universal Terminal System (UTS) 40 dumb terminals.

CAMS performs three general functions: 1) maintenance of the database, 2) retrieval of information from the database for local use, 3) reporting of data required by higher headquarters. The areas of CAMS that would be of primary interest to AOTS for interface purposes appear to be the Trainer Reporting Subsystem, Training Management Subsystem, Maintenance Personnel Subsystem, and the Personnel Related Records files in the database.

The Trainer Reporting Subsystem provides access to a database containing organization, equipment, status, and utilization data. This permits TRAINER work centers to update the appropriate data in the database in an on-line mode as events which affect trainers occur.

The Maintenance Personnel Subsystem provides the user the capability for monitoring manpower resources. Personnel data is maintained in the CAMS database and is accessible on-line. The system provides rapid access to administrative and personnel data pertaining to an individual or a group of individuals. Supervisors are able to update data pertaining to their personnel as changes occur. A variety of management products are provided to assist the administration activity in their personnel management.

The Training Management Subsystem provides the user the capability to forecast and schedule personnel training requirements. The types of training that can be monitored include recurring courses, one-time courses, on-the-job (OJT) training courses, certification courses, and upgrade training required by the Specialty Training Standards (STSs) and/or Job Qualification Standards (JQS). A variety of inquiries and reports are available on demand to assist supervisors in reviewing training requirements, scheduling training classes, and monitoring the progress of personnel in OJT and upgrade training.

The Personnel Related Records area of the database contains military personnel records and other employee related records. It also contains records related to job standards and time compliance technical order, and training related information.

(d) The Security Police Automated System. The Security Police Automated System (SPAS) is a database management system written in C Language. SPAS is comprised of two main subsystems: 1) the Personnel subsystem and 2) the Arms and Equipment subsystem. SPAS is currently hosted on standalone microcomputers and interfaces with users through pull down menus. Data related to personnel, manpower, and quality control, is currently loaded in manually. File transfers are accomplished through diskettes which are hand carried from machine to machine. The structure of the personnel data fields in the SPAS database appears to be identical to those in the PC III database.

Upgrades to SPAS include the networking of the standalone microcomputers through a LAN, which will also provide a communications link with a base level mainframe, and the interfacing of SPAS to PC III. PC III software will provide the interface capabilities.

(e) Advanced Training System. The Advanced Training System (ATS) will provide the Air Training Command with a technology based training system for each of the six Technical Training Centers (TTC). The program may also provide ATC with specifications for a system which is capable of exporting ATS courseware to the Field Training Detachments (FTD) and to AOTS training. The ATS could produce a functional specification for other major commands that is tailored to their particular training needs and environments. AOTS should benefit from ATS through increased widespread use of CBT courseware. Both government owned and commercial Computer Based Training (CBT) authoring systems should gain through increased use by both Air Force subject matter experts and contract courseware developers. The interfaces considered are management data and courseware. Batch exchange of management data may include the types of information included in ATR, OPTR, GPTR, and MTL files. CBT courseware may be routinely exchanged between ATS and AOTS.

3. System Backup and Restore. Large computer-based systems must allow for the automatic backup of data important to the system and the journaling of transactions. The prototype AOTS relies on the computer center backup procedures and did not implement embedded system backup and restore functions. The future AOTS would require the automatic backup of the system and the journaling of transactions.

C. Standalone Personal Computer Based Architecture. The standalone personal computer concept considers moving the full AOTS prototype program from the VAX 8650 mainframe computer and installing it on a personal computer. An example might be a Zenith Z-248.

In this scenario, the AOTS prototype will be modified to support 250 to 400 different AFSs, provide interfaces to other Air Force Systems, and include a journal and restore capability which were not included in the prototype. The software would be modified to interface with SQL and POSIX, and some comments would be added to the source code to improve maintainability.

The FSD program may deliver a software applications program on disk or tape and written documentation to the user. The user could then use the program to manage OJT. The using organization would develop or acquire MTLs, GPTRs, OPTRs, and training materials that are required in a specific implementation of the AOTS system.

Personnel data in the system may be periodically updated from AFMPC. This update might be via a magnetic tape, floppy disk, or modem communication. Personnel data must be updated quite often or the user will be frustrated with old data. A well-managed system would be updated possibly three times a week. To avoid AFMPC manually handling a myriad of requests from systems around the world, the data may not come directly from AFMPC, but from the servicing CBPO. The AOTS system administrator would receive the personnel data updates from the CBPO, and then run a program to install the new information in the ATR data base files.

OMC and CAMS data might be entered manually. Again, the data may be obtained in electronic format via tape, disk, or modem, but the system administrator would install the data. USAFOMC may be a logical source to compile MTLs. If this is so, the system administrator might receive a new MTL for an AFS periodically, perhaps semiannually, and install it.

In this scenario, multiple personal computers are available to users. The users can manage training, take training, or use office tools. An ad hoc report generator may be available to support the user. When a trainee enters the system, the system administrator may manually enter him or her into the system and insure that the new trainee is included in future personnel data updates. Upon leaving the organization, the trainee is provided with a hard copy and a disk of his/her ATR information.

1. Software Architecture. In this scenario, the software is separated from the hardware by the POSIX or other standard operating system interface. Provisions are made to install commercial applications software and office tools that operate through the POSIX interface to improve AOTS software independenc. from specific hardware and operating systems. This architecture is shown in Figure 2.

The data base is handled through the Structured Query Language interface. All applications software and users would then operate on data through the SQL which will aid portability among data base management systems. It is probable that applications software will be commercial off-the-shelf products.

The user gains access to the system by logging on through an access control. The user can use tools or applications software. The operating system is surrounded by the Portable Operating System Interface for Computer Environments (POSIX). The database management system would be accessed using the Structured Query Language (SQL) capability. The data base management system is relational in construction.

The computer software components is envisioned to be a combination of developed, Commercial off-the-shelf, and

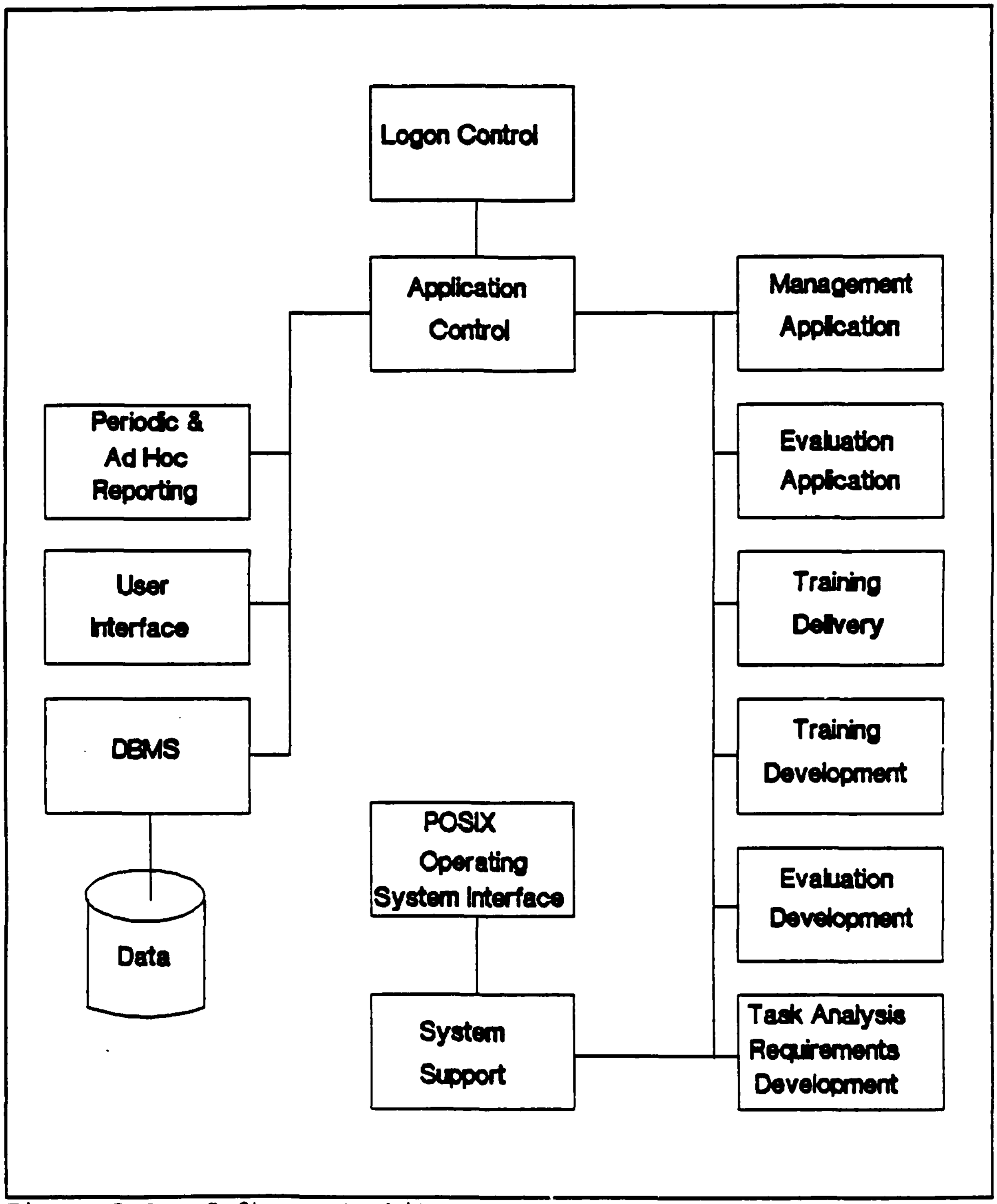

Figure B-2. Software Architecture, PC-Based System non-developmental as determined for the best interests of the Government.

The FSD AOTS software for the envisioned scenario would consist of one version. The software can be tailored when installed to fit the user's desired application and equipment. If the user only wants to deliver training, then the software would be installed to only perform that function. If the user has invested in an IVD delivery system, the software installation would activate the IVD delivery function.

(a) Operating System. The operating system is expected to be multi-tasking and surrounded by POSIX. The need to support multiple users and run long programs in the background, such as new MTL installation, is envisioned.

(b) Applications Software. Current DoD direction requires all contractor developed software to be written in Ada. Certain non-developmental software routines could be written in SQL, machine, or other languages.

1) POSIX. This is a standard operating system interface and environment that will support multi-tasking and encourage software portability at the source code level.

2) Database Management System. The relational database may require the use of the standard database query language, SQL, to support data base portability.

3) Tools. In this scenario, word-processing, spreadsheets, and ad hoc report generators are not envisioned to be integrated into the AOTS environment due to the limited capabilities of the PC machine.

2. Hardware Definition. In this scenario, the work center will provide the user with access to the system. Trainees, supervisors, developers, and managers at all levels may have access to terminals in the work area where they can manage, develop, evaluate or receive training. The work center is the interactive input/output station for the AOTS.

The terminal configuration at the work center may be supported with a family of peripherals that can be added to in a modular manner to meet the needs and budget of the user. The user can install multiple terminals. Equipment may be available to deliver both IVD and CAI lessons. A selection of printers would be available to support management analyses, notices, reports, and testing materials. Optical mark readers may be offered for work centers to score off-line tests and perhaps enter performance evaluation results information. The future design could be modified to accommodate other peripherals which include light pen, joy stick, mouse, and digitizer. The work center will communicate with other AOTS systems in this scenario through a manual interface. The system administrator can transfer files via modem.

Interactive Video Disk courseware may be resident on the work center terminal. The microprocessor used with the IVD player present the free standing IVD lesson delivery.

The work center software will perform the full range of AOTS functions. The range of functions may cover such areas as edit OPTRs, ATRs, and ITRs. The work center software also performs qualification assessments, generates on-line and off-line tests, directs quality control evaluations, and produces reports. Data stores include ATRs, ITRs, GPTRs, MTLs and the test item bank.

This range of work center software functions is extensive and requires large computer resources to handle all functions concurrently. The functions of the AOTS will be trimmed to fit within the resources of the PC, or the functions will be partitioned and a subset of the total functionality will be handled at a time.

D. Networked System Architecture. Networking the PC based systems described in the previous section is another viable architecture for FSD AOTS. Networking allows the apportioning of the AOTS program files and functions among a hierarchy of processors which yields a more automatic system and provides the ability to design in management reporting functions.

1. Hardware Definition. Hardware architecture can be envisioned as occurring in five hierarchial levels connected with a communications system and is shown in Figure B-3.

(a) work center (Level I). The work center would provide the user with access to the system. Trainees, supervisors, developers, and managers at all levels would have access to terminals in the work area where they could develop, manage, and receive training. The work center is the interactive input/output station for the AOTS. Multiple users can be supported by the work center CPU.

The terminal configuration at the work center may be supported with a family of peripherals that can be added to the terminal configuration to meet the needs and budget of the work center terminal user. Equipment could be available to deliver both IVD and CAI lessons. A selection of printers may be available to support word processing, management analyses, notices, reports,

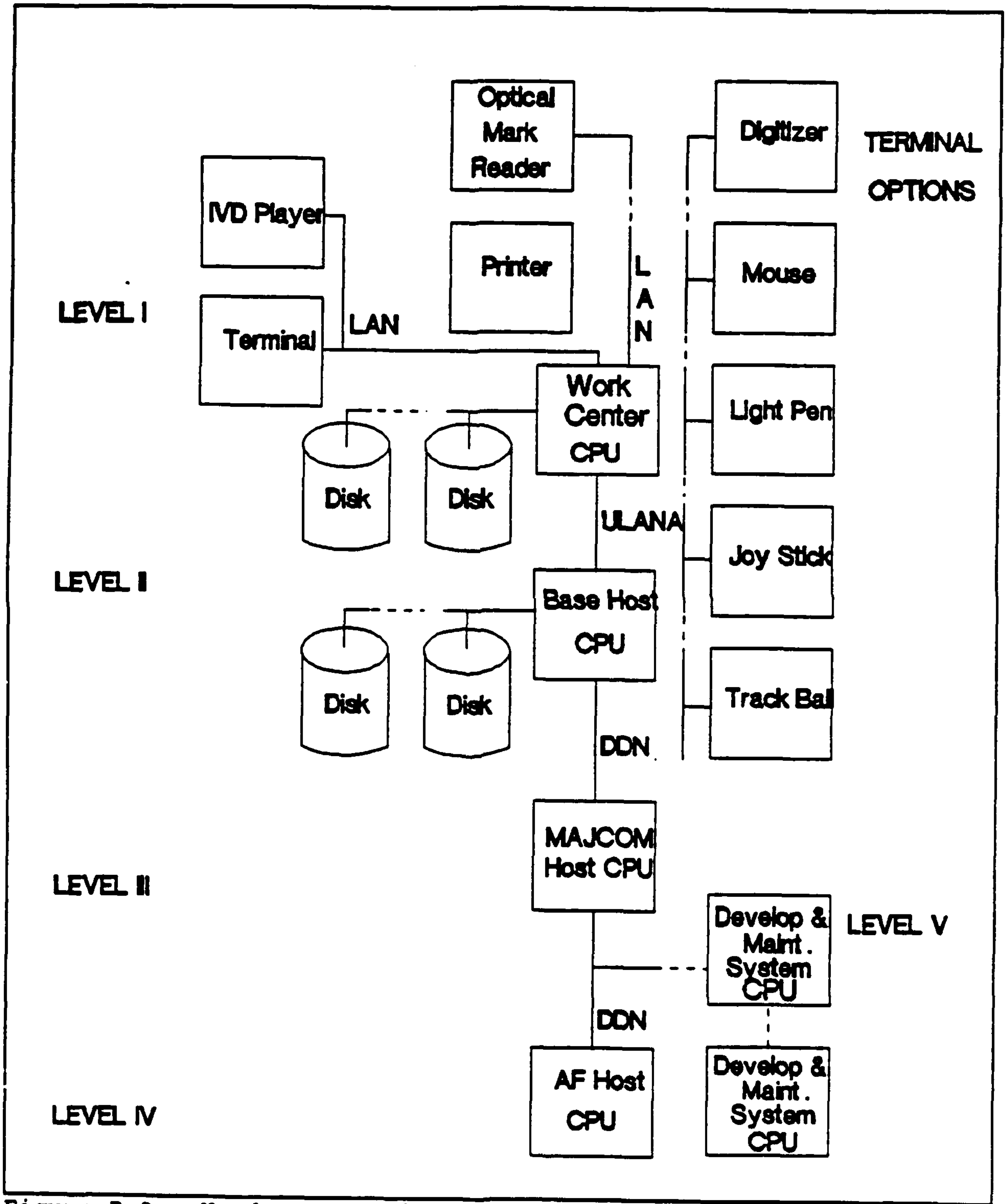

Figure B-3. Hardware Architecture, Networked System and testing materials. Optical mark readers may be offered to work centers to score off-line tests and perhaps enter performance evaluation results information. Other peripherals include light pen, joy stick, mouse, and digitizer. The work center will communicate with the Base Host via a local area net in this scenario.

Interactive Video Display courseware may be resident on the work center terminal. The microprocessor used with the IVD player present the free standing IVD lesson delivery.

The work center performs as file server and supports all local file management. The work center can support multiple terminals and communicates with the base host via ULANA.

The work center applications software edits OPTRs, ATRs, and ITRs. Resident in the level I database are OPTRs. The work center software also performs qualification assessments and contains office tools.

(b) Base Host (level II). The Base Host communicates with other levels by ULANA and DDN. Main storage is expected to be large and expandable to handle the considerable data storage requirements for this level. The base computer can support a group of geographically and functionally related users.

The Base Host software generates on-line and off-line tests and directs Quality Control (QC) evaluations. Level II software generates all reports such as QC reports and system evaluation reports. Office tools that were purchased off-the-shelf are integrated into the AOTS and resident on the Base Host as at other levels. Level II data stores include ATRs, ITRs, GPTRs, MTLs, and the test item bank. Tentative Master Task Lists (TMTLs) are working copies of MTLs that are modified to include tasks unique to that user.

(c) MAJCOM Host (Level III). This level provides management capability to responsible personnel at the headquarters level. MAJCOM Host is capable of supporting base hosts under it. It would control Level I systems at facilities that have a very limited AOTS requirement that does not justify the investment in a Level II system. The MAJCOM is capable of communicating with ULANA and DDN. The MAJCOM requires at least one terminal for interacting with AOTS.

(d) Air Force Host (Level IV). The Air Force Host provides overall Air Force management of the AOTS system. It can communicate with all Level II and Level III systems to gather management information. It requires at least one terminal for interaction with AOTS.

(e) Development and Maintenance System (Level V). The AOTS system may require one or more sites that develop courseware and provide system maintenance functions. Recognizing that any engineered system requires maintenance and updating during its useful life, the AOTS will require a support organization to evaluate necessary software and hardware design changes that must occur. The maintenance organization would have engineering responsibility to include issuing standards, operating instructions, configuration management, and other system engineering areas.

(f) System Interfaces. AFMPC data may be periodically used to update the base host with personnel data via a magnetic tape or modem. USAFOMC and CAMS data may be entered manually.

(g) User Interfaces. The future AOTS system may consider supporting windowing and pop-up menus. Pointing devices such as mouse, light pen, and trackball could be included in the design for user convenience. Structured Query Language (SQL) is expected to be specified to communicate with the data base. An ad hoc report generator and office tools may support the user.

2. Software Architecture. The pictorial view of the software architecture is shown in Figure B-4. The hardware is separated from the software by the POSIX and DBMS interfaces. All applications software and office tools would communicate with the operating system through the POSIX interface which will improve AOTS independence from specific hardware and operating systems.

The data base is handled through the SQL interface. All applications software and users may be required to operate on data through the SQL which will aid portability among data base management systems. It is envisioned that commercial off-the-shelf products would provide the data base management system.

The user gains access to the system by logging on through an access control. The user can use tools or applications software. The operating system is surrounded by the Portable Operating System Interface for Computer Environments (POSIX). The database management system would be accessed using the SQL capability. The data base management system is relational in construction.

The FSD AOTS software is expected to consist of one version at all levels of implementation. The computer software

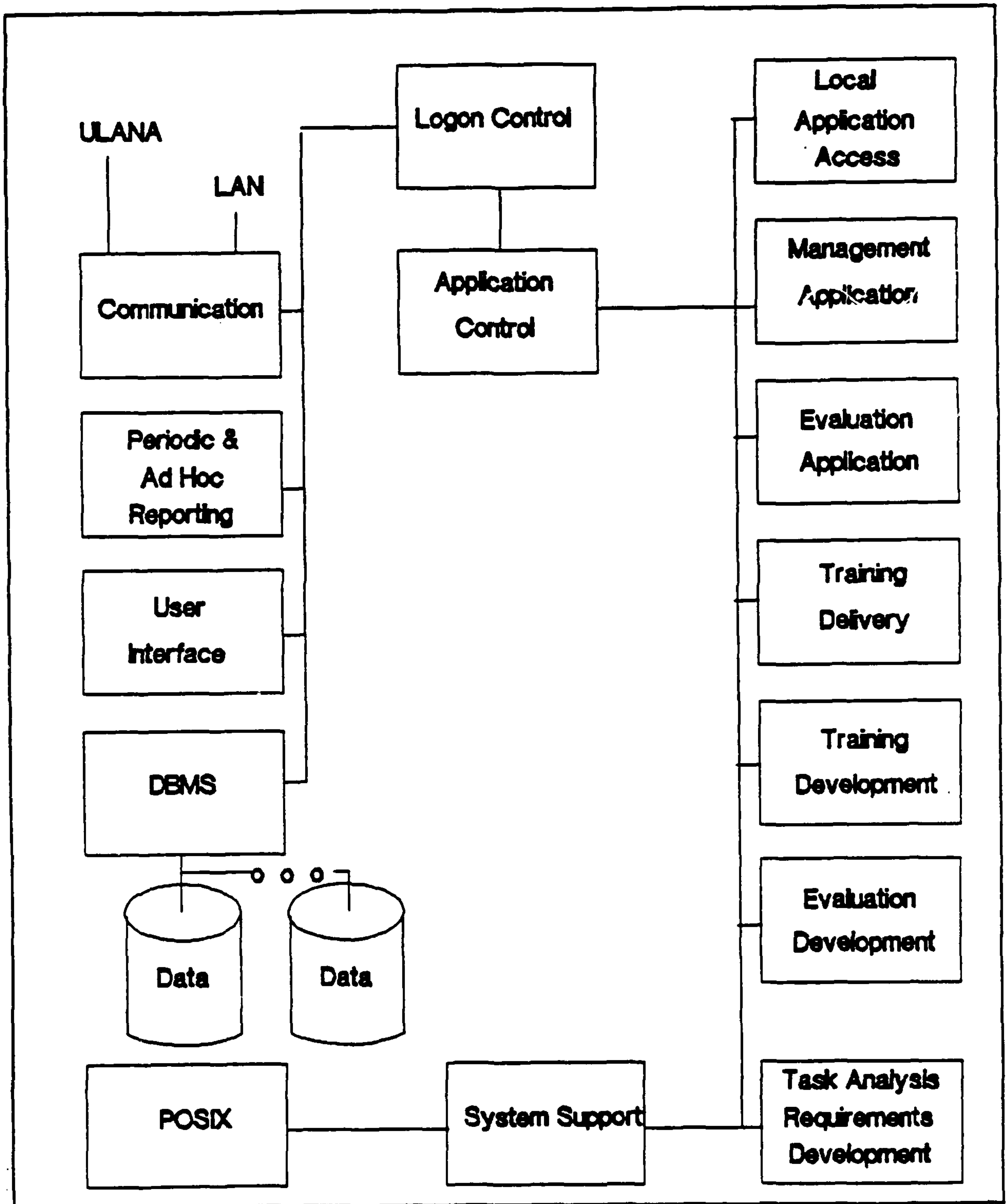

Figure B-4. Software Architecture, Networked System components could be a combination of developed, Commercial Off-The-Shelf (COTS), and non-developmental as determined for the best interests of the Government.

(a) Operating System. The operating system may be multi-tasking and surrounded by POSIX.

(b) Applications Software. Contractor developed software may be written in Ada. Certain non developmental software routines could be written in SQL, machine, or other languages.

(1) POSIX. This is a standard operating system interface and environment that will support multi-user and encourage software portability at the source code level.

(2) Database Management System. The relational database may be required to use the database query language, SQL, to support data base portability.

(3) Tools. Word-processing, spreadsheets, and ad hoc report generators could be purchased COTS and integrated into the AOTS environment.

E. Supportability Enhancements. The prototype AOTS software has a number of supportability shortcomings which are likely to have a negative effect on the life cycle cost of re-using the prototype software. In particular the prototype configuration management (CM) system, test and evaluation (T&E) program, and software maintenance tools need to be improved. There is no direct AOTS software modification required for these improvement, but the effort required to implement new CM, T&E, and tools is still significant. The decision on which approach to producing an FSD AOTS is most cost effective should take the cost of improving prototype support ability into account.

1. Configuration Management Program. The future AOTS user may choose to establish a production-quality configuration management (CM) program. An effective configuration control and product quality assurance program is needed for a supportable production program. With AOTS systems dispersed possibly around the world, the entity with engineering responsibility for that system must know what the exact structure of the software and hardware is to correct apparent bugs and design deficiencies.

The task to write product specifications can be strongly aided by a software program. The product baseline must be carefully defined and controlled. A CM library should be established. A change system would be established that will insure that design changes get full and careful consideration prior to approval and provide an audit trail for modifications.

2. Test and Evaluation Program. A production-quality test and evaluation program will be important to field a robust production system. The future AOTS builder will desire to catch as many code errors as possible before distributing software around the world.

Test cases would be developed and incorporated into the production specification. Test procedures would then be updated with the test cases. Quantitative and measurable pass/fail criteria would be established for each recorded result.

3. Software Maintenance Tools. The future AOTS program may desire a language-oriented programming environment where all programs are written in Ada. This means that there will not be a need for a general-purpose program editor to create and edit source code. A special purposes editor would be used with built-in knowledge of the language.

The environment may include computer aided software maintenance and program management tools. It is generally accepted that a software program is fully specified by data flow diagrams and data dictionaries.

The production program will have the task to develop and control production-quality design documentation to aid cost effective software maintenance. This documentation task includes the automated development of data flow diagrams, data dictionaries, and similar documentation. The tools would be controlled with configuration management system so documentation will remain current. Another task envisioned for the future program would be to write all procedures necessary to operate and maintain AOTS. The task would include such areas as installing MTLs and TMTLs, system administrator duties, compilation orders, etc. The future program may desire to manually write data definitions since the syntax of Ada does not include this information in the code.

SECTION II - ESTIMATED CHANGE REQUIREMENTS

A. Prototype Enhancements.

1. Software Program Characteristics. Significant components of the Instructional Support System (ISS) computer program are integrated into the AOTS prototype. The ISS and AOTS programs overlap and are used together to provide the complete set of functions that are included in the AOTS definition. AOTS incorporates some ISS source code and uses ISS programs to develop and deliver lessons and provide system utilities.

At this point in the software life cycle, there is no technical difference between ISS and AOTS prototype computer programs. During the prototype development program, they were treated as separate entities for the purpose of management and software maintenance. It is assumed that any future user of the AOTS prototype could appropriate whatever ISS source code is needed and encompass it and maintain it within the reused system. There is no need for distinction between ISS and prototype AOTS for the software engineer. The ISS source code and the AOTS prototype source code may all be adopted by the future AOTS prototype program user.

It is generally accepted that if a package needs more than 50% rewrite, then is should be redone from scratch, i.e., it is the same as a 100% rewrite. In many instances a 25% rewrite is less life-cycle-cost-effective than a 100% re-write.

For the purpose of analysis, the software source code for the prototype is divided functionally into nine Computer Program Components (CPC). There was no effort to make the CPCs correspond to the Configuration Items that are delivered under the prototype development contract. These nine CPCs are Management Application, Evaluation Application, Training Delivery, Training Development, Evaluation Development, Task Analysis Requirements Development, Virtual Machine Layer, System Support, and General Support.

For two of the CPCs, the metrics summary at Appendix A show that Training Development and Training Delivery are high level CPCs. Training Delivery consists totally of ISS packages, and analysis shows that The Training Delivery Computer Program Component (CPC) contains 46,709 Source Lines of Code (SLOC).

Four of the CPCs are similar to each other in that they are in general high level packages that perform interactive user functions. They are: 1) Management, 2) Evaluation Application, 3) Evaluation Development, and 4) Task Analysis Requirements Development Applications CPCs. The metrics summary in Appendix A shows that from 19.7 to 22.5% of their statement implement on-line communications. On-line communications includes such things as input from the keyboard and printing. The percentage of statement types in these packages closely correlate in terms of statement characteristics.

The last three CPCs are at or close to the machine layer. The Virtual Machine Layer (VML) CPC is designed to contain all machine dependencies. System Support is a low level CPC that is close to the VML. Some of this code is written in VMS assembly language which is uniquely designed for the VAX line of computers. General Support CPC consists of about 3,600 lines of which 551 are Ada, and 3,053 are SAS. SAS is the Statistical Applications for the Social Sciences language that is an application package available for many main-frame machines.

2. Generalize to Work for Other AFSs. A basic requirement for the reuse or porting of the prototype AOTS is the expansion of the prototype capability from its present four Air Force Specialty Codes to a much larger number. First, a simple AFS change is considered for the prototype and then that beginning is enhanced in Section II.B.3 to include the case where prototype software is ported to another processor. In Section II.B.3, the code is modified for the SQL and POSIX or other software standard interface and all non-Ada code would be rewritten.

At first, the scenario considers the modification of the present prototype installed on the VAX 8650 to accommodate 250-400 Air Force Specialty codes. Other than the AFS expansion, the prototype code contains the same functionality and capability as the prototype. Further, the code is not modified for the SQL and POSIX interface and comment lines are not added to the code to improve understandability.

Currently the prototype creates and manipulates 13 AFS-unique data files for each AFS. Each user of the prototype is identified by one of the four supported AFSs. The software uses that identified AFS to process AFS-unique actions for that user. Within the AOTS program file structure, a numbering convention can be devised for AFS-related files based on the unique AFS number and directories can be established to hold them.

No effort has been expended to streamline the MTL installation process. It is reasonable to expect that an investigation and re-design of the MTL files and installation processes would result in significant economies in file space and installation time.

Today, many AOTS reports include information on all four currently supported AFSs. If the system were supporting in the area of 250 to 400 AFSs, then reports must be re-designed precisely for the expanded use. Significant re-design will be made in the report portion of the AOTS prototype to satisfactorily service 250 or more AFSs in a single system.

The discussion that follows outlines the effort to generalize AFSs. The existing code was viewed. The code in each Computer Program Component (CPC) was examined to determine to what extent it must be changed to generalize the AFSs.

The prototype code was examined to consider expansion of the AFS capability from the present capability for four to as many as 400. The Management, Task Analysis and Requirement Development, and System Support CPCs are affected, with the most change occurring in the Task Analysis and Requirement Development CPC.

AFS-unique files are accessed through the code unit named "AFSC". AFSC contains 157 SLOC. The package contains global variables that calling packages use to manipulate AFS-unique files. Currently, the package exhaustively names all files for the four AFSs and uses a case statement to select the AFS to manipulate. This package will be re-designed 100% to allow variable-length loops to process AFSs. File name convention will be devised to allow the AFS number to be converted logically into file names.

All packages that manipulate AFS-unique data contain the package named "AFSC" in their context statements. Sixty-six other units call AFSC through the "with" context statement. A survey of these identified six that will require code changes. AOTS_LOGON will require two re-designed procedures to eliminate hardwiring. FINAL_MTL_GEN will require additional functions to implement MTLs in groups, in addition to the options to install one AFS and install all AFSs. ITEM will be partially re-designed to add loops to handle the increased number of AFSs. MTL_BUFFER uses an enumeration type for the AFSs that will be re-designed. MTL_FUNCTION_SELECT requires increased functionality. PDS_TO_AOTS will be modified to allow more options for components and AFSs. Functions must be added to allow the user to selectively list AFSs to be set up during MTL installations.

Screens would be re-designed to allow paging through AFSs instead of the single screen display that is satisfactory now for displaying four AFSs. Reports would be significantly reformatted to provide useful outputs tailored to the AFS and user's needs. A summary of the changes required to expand AFSs is included in Table B-1.

Table B-1. Line Changes Required to Generalize AFSCs.

CPC NAME	SLOC	CHANGED LINES
Management	15,734	200
Evaluation Development	4,269	85
Task Analysis Requirement Development	18,554	786
System Support	15,290	100
General Support	3,604	634
Total Changed Lines:		1,805

3. Porting the Modified Prototype Software. In practice, any program other than a trivial one requires some modification when it is moved from one machine platform to another. These modifications come about through inconsistencies in compilers, operating system dependencies, and machine architecture differences.

This section considers the case where prototype software is being ported to another processor, but the code still contains the same functionality and capability as the prototype. The code is modified for the SQL and POSIX or other software standard interface. Some comment lines are added to the code to improve understandability. It assumes all non-Ada code would be rewritten.

A good estimate of SLOC that require rewrite can be obtained by a line-by-line analysis of the source code. However, if the resources are not available for such an exhaustive bottoms-up approach, a parametric approach is appropriate that looks at characteristics of the code and compares findings on similar analyses with the case at hand to draw parallels. This analysis uses the parametric approach and complements it with a bottoms-up estimates for very limited portions of the code.

The existing code was grouped into Computer Program Components (CPCs) and the function of each CPC was viewed. If it provided functionality that could be used in the porting to the new machine, the CPC was considered for porting. The CPCs were judged for required modifications by evaluating the following statements: The code is hardware independent; The comments are meaningful and complete; The code is written for the SQL interface to the DBMS and the POSIX or another standard operating system interface.

To the extent that the code met these criteria, the number of rewritten and new Source Lines of Code (SLOC) was estimated. Each CPC was examined to determine to what extent it meets the criteria and a number of changed lines was estimated.

Implicit in the changed lines estimate is the assumption that the prototype software works well enough. Software Problem Reports that exist on the prototype software discuss un-documented features along with definite software "bugs". The future user will desire to evaluate the impact of these reports on the functionality that the FSD AOTS will specify.

A recent analysis of the ISS SLOC used a 100% line count (Computer Systems Security Research, Development, Assessment and Technology Transition II - Evaluation of ISS for Use in ATS, Final Report, January 29, 1988, prepared for USDOT). The line count data provided useful information to scope the rewrite of SLOC for a scenario to port the ISS code in this case. The ISS computer program is analogous to the AOTS prototype in many ways.

ISS and AOTS SLOC overlap and are used together. AOTS incorporates some ISS source code and uses ISS programs to develop and deliver lessons and provide system utilities. At this point, there is no difference between ISS and AOTS prototype. During the prototype development program they were separate entities for the purpose of management and assigning responsibility for software maintenance. Any future user of the AOTS prototype would take whatever ISS code is needed and encompass it and maintain it within the reused system. There will be no distinction between ISS and prototype AOTS. It would all become the ported AOTS software program.

ISS and the prototype AOTS were designed and written by the same development team and reside in the same software system. The recent ISS analysis counted roughly 266,000 lines of code and because of the similarity of the work, we can apply some of those findings to our case.

The recent ISS analysis found that almost all of the code was contained in packages that needed from zero to 50% changed lines. If a package needed more than 50% rewrite, then it should be recoded from scratch, i.e., a 100% rewrite. The ISS findings fit roughly into Table B-2. The first column is the percent of code to be changed in a given package. The second column is approximately the total lines of code that reside in packages needing that percent of change. Column three is the product of the first two. For example, 2000 lines of code reside in packages that must be rewritten 5%.

Table B-2. Characteristics of Development and Delivery Code.

Percent Change Required	Lines of Code	Change Lines
0%	10,000	0
5	2,000	100
10	26,000	2,600
15	21,000	3,150
20	97,000	19,400
30	39,000	11,700
40	50,000	20,000
50	11,000	5,500
100	10,000	10,000

The changed lines represent about 27% of the total lines of code. Which means that approximately 27% of the total ISS lines that were evaluated would have to be rewritten in the porting scenario.

Training Delivery consists totally of ISS packages and was counted 100%. A hand count of the data shows that 22.5% of the Training Delivery CPC would be rewritten.

Training Development is very similar in characteristics Training Delivery; however, the recent ISS analysis did not count it completely because some of the source code units were written expressly for AOTS. Of the lines that the analysis did count, the change percentage corresponded closely to the 22.5% change computed for Training Delivery. Regardless of the fact that the analysis did not count all of these code files, similarities between Training Development and Training Delivery are significant.

The metrics summary at Appendix A show that Training Development and Training Delivery are high level Computer Program Components (CPCs) and should not need as many changes as the CPCs closer to the machine layer. The calculations for the analysis will use 22.5% as the lines needing change in the higher level CPCs.

Management, Evaluation, Evaluation Development, and Task Analysis Requirements Development Applications CPCs are similar to each other because they are in general high level packages that perform interactive user functions. The metrics summary in Appendix B shows that from 19.7 to 22.5% (which is a high percentage considering most statements are branching logic, declarations, and assignments) of their statement implement on-line communications which includes such items as input from the keyboard and printing. These CPCs would need 22.5% change. Also, the Task Analysis Requirements Development CPC contains 2% DCL commands that would be rewritten.

Virtual Machine Layer (VML) software is designed to contain all machine dependencies. Since this code is largely machine dependent the VML must be rewritten 100%. SLOC for this CPC is estimated to be 6980. Comment lines are not counted as SLOC anywhere in this analysis.

System Support is a low level CPC that is close to the VML and would probably require significant change. Some of this code is written in VMS assembly and would have to be rewritten. The recent ISS analysis showed that very little Ada code is rewritten over 50%. Conservatively, a 50% rewrite would be required for this CPC.

General Support CPC consists of about 3,600 lines of which 3,053 are SAS. 22.5% Ada rewrite plus 100% SAS translation means that 3,343 SLOC would be written/rewritten for this CPC.

It is desired to expand the above scenario to allow for expansion of the AFSs that the system can handle from the present capability of four to as many as 400. AFS-unique files are accessed through the code unit named AFS as previously discussed. Screens would be designed to allow paging through AFSs instead of the single screen display that is satisfactory now for the four AFSs.

The AFS capability expansion would not add to the Re-work of General Support and System Support. Reports would be reformatted, but they are already being rewritten 100% because they are written in SAS. The new formats would be included in the new design and this significant rework effort will not add to the total rewritten SLOC. System Support is also experiencing a significant re-write. The AFS expansion within System Support can be handled within the budgeted 50% re-write. The Management, Task Analysis and Requirement Development, and System Support CPCs are affected most by the increased AFS capability requirement, with the most change occurring in the Task Analysis and Requirement Development CPC.

Table B-3. SLOC Changes Required to Port to Mainframe

CPC NAME	SLOC	CHANGED LINES
Management	15,734	3,740
Evaluation	5,568	1,338
Training Delivery	46,709	10,509
Training Development	99,747	22,443
Evaluation Development	4,269	960
Task Analysis Requirement Devel.	18,554	5,161
Virtual Machine Layer	6,975	6,975
System Support	15,290	7,645
General Support	3,604	3,342
Total:	216,450	62,115

Percent Change: 29%

Table B-3 is a summary of the SLOC changes based on the preceding discussion.

4. Interface to Other Air Force Systems. This paragraph identifies the changes and the effort required to provide the AOTS software with interfaces to other Air Force systems. The change estimates include such topics as types of changes required to implement the interface, Source Lines of Code (SLOC) change estimate, and identify technical risk areas.

(a) Personnel Concept III. Personnel Concept III (PC III) is a large computer-based system that decentralizes the processing of personnel data to the unit level through the use of remote terminal work centers in unit orderly rooms connected to a base-level mainframe. The unit-level work centers are staffed by personnel specialists and perform approximately 64 personnel functions that used to be performed at base CBPOs.

The Automated Personnel Data System II (APDS II) system is hosted on a base-level Phase IV Sperry (UNISYS) 1100/60 mainframe. The PC III system will be hosted on a UNIX baase 3B2 minicomputer. A communications network will connect the base computer to other Air Force facilities.

Possible interfaces between AOTS and PC III include personnel data, manpower data, electronic mail, education data, and printers and other peripherals.

PC III data including personnel, manpower, and education data, will be used within the AOTS, but no transmission of these data from AOTS to PC III is envisioned. In order to control the update of these data, the Air Force will probably not allow updates from individual work centers. Software can be written to transfer these PC III data to AOTS files. In general, the process requires an understanding of the PC III and AOTS file structures, and how the data flows through the software on the respective machines. Then an algorithm can be designed and coded to convert the electronic files from one format to the other.

These data are characterized by ASCII or binary files and contain mathematical and textual information. The source lines of code estimate is small because it involves reading data, then manipulating data, and finally writing to files. There are no significant timing constraints. The process can be run off-line in batch and use relatively simple algorithms. Technical risk that the job can not be accomplished within a budget and schedule is low.

A conversion is routinely accomplished now to transfer APDS data to the AOTS to update the personnel portion of the ATR files. The conversion program contains 300 source lines of code. The prototype conversion program received input from a tape and runs during down hours. Three hundred lines is a reasonable estimate for the free standing PC scenario that is discussed by this plan. A more general implementation for the production AOTS for the mainframes and networked scenarios is estimated at 1000 lines of Ada source code. The thousand line estimate would allow for the automatic rendezvous and receipt of information over a long haul communication link such as DDN.

Electronic mail and the sharing of printers and other peripherals have not been mentioned as requirements for the FSD AOTS, but they become possibilities with the future AOTS electronically connected with the PC III node in the work place.

(b) Air Force Occupational Measurement Center. The USAF Occupational Measurement Center (USAFOMC) has developed occupational data bases for all specialties within the Air Force that have 200 or more incumbents. One function of the occupational analysis program is the development of comprehensive task lists that include all tasks required to be performed in a job specialty as well as all tasks that are performed but are not prescribed by regulations or manuals. These definitions of tasks as used in occupational analysis are substantially more specific and granular than found in AFR 39-1 or the STS. It is not clear at this time what amount of USAFOMC information is desired by the FSD AOTS.

(c) Core Automated Maintenance System. The Core Automated Maintenance System (CAMS) is a large, real-time, on-line system used at the base-an d unit-level to manage maintenance equipment and personnel resources. CAMS also provides much of the maintenance data needed by major commands, Air Force Logistics Command, Headquarters USAF, and other agencies to manage and track maintenance resources worldwide. The system applies to aircraft, missile, and Communication-Electronics (C-E) maintenance.

CAMS provides the capability for maintenance personnel to communicate to a central base-level computer via remote terminals in selected maintenance work areas. CAMS is designed to operate on the standard base-level Phase IV x2/x3 configuration using Sperry (UNISYS) 1100/60 Computer Systems. The remote terminals emulate Sperry Universal Terminal System (UTS) 40 dumb terminals.

CAMS performs three general functions: 1) maintenance of the database, 2) retrieval of information from the database for local use, and 3) reporting of data required by higher headquarters. The CAMS database is an integrated database that consists of ten files. Within each CAMS area there are a number of record types defined. The basic files are: 1) Major Control Records, 2) Isolated CALC Entry Point Records, 3) Personnel Related Records, 4) Event Related Data, 5) Equipment Related Data, 6) Reject Codes and Narrative Records, 7) Area Records, 8) Status Related Data, 9) Maintenance Data Collection History, and 10) Transaction Data.

Local managers can control their resources through the use of 17 CAMS subsystems. These are: 1) Comprehensive Engine Management System, 2) Test Equipment Reporting and Management, 3) C-E Equipment Status and Inventory Reporting, 4) Maintenance Events, 5) Trainer Reporting, 6) Location, 7) Data Location, 8) Inquiries, 9) Status and Inventory Reporting, 10) Operational Events, 11) Inspection and Time Change, 12) Equipment Transfer Procedures, 13) Time Compliance Technical Order, 14) Maintenance Personnel, 15) Training Management, 16) Personnel Transfer Procedures, and 17) Configuration Status Accounting System (CSAS).

The areas of CAMS that would be of primary interest to AOTS for interface purposes appear to be the Trainer Reporting Subsystem, Training Management Subsystem, Maintenance Personnel Subsystem, and the Personnel Related Records files in the database.

The Trainer Reporting Subsystem provides access to a database containing organization, equipment, status, and utilization data. This permits TRAINER work centers to update the appropriate data in the database in an on-line mode as events which affect trainers occur.

The Maintenance Personnel Subsystem provides the user the capability for monitoring manpower resources. Personnel data is maintained in the CAMS database and is accessible on-line. The system provides rapid access to administrative and personnel data pertaining to an individual or a group of individuals. Supervisors are able to update data pertaining to their personnel as changes occur. A variety of management products are provided to assist the administration activity in their personnel management.

The Training Management Subsystem provides the user the capability to forecast and schedule personnel training requirements. The types of training that can be monitored include recurring courses, one-time courses, on-the-job (OJT) training courses, certification courses, and upgrade training required by the Specialty Training Standards (STSs) and/or Job Qualification Standards (JQS). A variety of inquiries and reports are available on demand to assist supervisors in reviewing training requirements, scheduling training classes, and monitoring the progress of personnel in OJT and upgrade training.

The Personnel Related Records area of the database contains military personnel records and other employee related records. It also contains records related to job standards and time compliance technical order, and training related information.

The CAMS database can provide AOTS with resource data to track resource availability for training events. The CAMS scheduling data can be used by AOTS to match training events up against operational maintenance actions. Both CAMS and AOTS contain position qualification and certification data that can be shared.

CAMS data could be used within the AOTS and transmission of data from AOTS to CAMS is also envisioned. Software can be written to transfer CAMS data to AOTS files and visa versa. In general, the process requires an understanding of the CAMS and AOTS file structures and how the data flows through the software on the respective programs. Then an algorithm can be designed and coded to convert the electronic files from one format to the other.

These data are characterized by ASCII or binary files and contain mathematical and textual information. The source lines of code estimate is small because it involves reading data, then manipulating data, and finally writing to files. There are no significant timing constraints. The process can be run off-line in batch and use relatively simple algorithms. Technical risk that the job can not be accomplished within budget and schedule is low.

As previously mentioned, a conversion is routinely accomplished now to transfer APDS data to the AOTS to update the personnel portion of the ATR files. The conversion program contains 300 source lines of code. The prototype conversion program received input from a tape and runs during down hours. Three hundred lines is a reasonable estimate for the free standing PC scenario that is discussed by this plan. A more general implementation for the production AOTS for the mainframes and networked scenarios is estimated at 1000 lines of Ada source code. The thousand line estimate would allow for the automatic rendezvous and receipt of information over a long haul communication link such as DDN.

(d) The Security Police Automated System. The Security Police Automated System (SPAS) is a database management system written in C Language. SPAS is comprised of two main subsystems: the Personnel subsystem and the Arms and Equipment subsystem. SPAS is currently hosted on Z-248 and Z-100 stand-alone microcomputers and interfaces with users through pull down menus. Data is currently loaded in manually. File transfers are accomplished through diskettes which are hand carried from machine to machine. The structure of the personnel data fields in the SPAS database are identical to those in the PC III database.

Planned upgrades to SPAS include the networking of the standalone microcomputers through a LAN, which will also provide a communications link with a base level mainframe and the interfacing of SPAS to PC III. PC III software will provide the interface capabilities.

The future SPAS hardware architecture contains the basic elements of the FSD AOTS networked scenario. Because of the future link to PC III and the networked nature of the system, there are many opportunities for interfaces between SPAS and AOTS. Personnel data, manpower data, Quality Control data, electronic mail, and CAI instruction are possibilities. While these interfaces are possible, and no interface between AOTS and SPAS are envisioned.

(e) Advanced Training System. The Advanced Training System (ATS) will provide the Air Training Command with a technology based training system for each of the six Technical Training Centers (TTC). The program will also provide ATC with specifications for a system which is capable of exporting ATS courseware to the Field Training Detachments (FTDs) and to on-the-job training. The ATS will produce a functional specification for other major commands that is tailored to their particular training needs and environments.

Computer Based Training (CBT) will be required in the 1990's and beyond, due to the rapid increase in weapon system complexity. The ATS is envisioned to provide the technology required to supplement the limited instructor personnel and to cope with the more complicated systems. Poth Government-owned and commercial CBT authoring systems would benefit through increased widespread use by both Air Force subject matter experts and contract courseware developers. Depending on the degree of implementation, the entire appearance and operation of training in the Air Force could change toward the use of CBT. The ATC estimates that the ATS could save as much as $150 million in operating costs over the life cycle of the system, but the emphasis is on the improved quality of the training and the better management of that training.

For the ATS, two types of data interface are considered. exchange of management data and courseware.

(1) Management data. Batch exchange of management data such as ATRs and ITRs could be routinely handled. Once it is decided what data is to be transferred to ATS, AOTS programs would be required to open the database and select the data to be sent. The algorithm may be complicated somewhat by searching, sorting, and merging data from many different files. The transferred data may be formatted as ASCII files to avoid machine differences between AOTS and ATS. If information is to be transferred from ATS to AOTS, programs would be needed to receive the file, parse it, and store the data. Percent lines of code is small for any file transfer scenario, because it involves a straight-forward algorithm of reading data, then manipulating data, and finally writing to files. There are no timing constraints, the work can be done off-line in batch, and uses simple algorithms. Technical risk is low.

(2) Interchangeable courseware. The ATS program is expected to set the standard for Air Force CAI course-

ware. The FSD AOTS program schedule will follow the ATS program in time. The ATS entered full scale engineering development in April 1989. If the FSD AOTS project was ordered to proceed today, it would trail ATS by at least a year. One year is an optimistic schedule for an Air Force project office to conceptualize, request proposals, and award a contract for the FSD AOTS. A logical programmatic assumption is that the ATS specify CAI courseware first and will set the de facto standard for courseware among Air Force CAI systems.

ATS courseware will be usable on AOTS if the ATS establishes a written specification for courseware. If the ATS poorly defines the courseware specifications or capriciously changes them, the exchange of courseware would not routinely occur.

There is low technical risk to ATS courseware compatibility if ATS establishes the courseware specifications before the ATS enters FSD. As the AOTS continues through development without a satisfactory development specification the risk to courseware compatibility climbs steeply. As both AOTS and ATS make commitments to specific hardware and software formats the risk increases that the compatibility task will be able to be accomplished for a given cost.

Use of ATS courseware on AOTS will require effort because the hardware and software will, in the general case, be incompatible. But ATS specifications could include requirements that would aid courseware transportability, such as specifying a hardware-neutral scale for graphics.

Software can be written to transform ATS courseware to AOTS. In general, the process requires an understanding of the ATS and AOTS courseware file structures, and how the data flows through the course on the respective machines. Then an algorithm can be designed and coded to convert the electronic files from one course format to the other.

A conversion was recently accomplished to move ISS courseware from the VAX and presentation on the Tektronix 4105 terminal to the Z-248 PC. Both were running ISS. The conversion program was estimated at 1000 lines of Ada source code.

(f) Distributing the Prototype. In the prototype AOTS implementation the processing power of the Z-248 was largely unused. The Z-248 was configured to perform as a "dumb" terminal. From a cost point of view, this was certainly not incorrect because the cost to the project of the Air Force standard Z-248 was less than a terminal. However, it would be a more efficient use of processing power if a design could be arrived at that would exploit the processing power and memory of the Z-248 and off-load some of the tasks assigned to the VAX 8650 as in a true distributed system.

In a distributed system, two processors or more work together on the same program and share memory. An example for the prototype AOTS might be to use the Z-248 to store output for the associated work center printer. Currently, the printers must run continually whenever the AOTS is running or they lose print files sent to them when they are off-line. The ALPS 2000 printers are not designed for continuous non-attended operation. A paper jam could lose the whole night's output to the work center. If the output were spooled on the Z-248, the printing could be done on command by when the user is present to monitor the printer.

One disadvantage of a distributed program is communication channel delay. An Ada programmer may access any global object through the "with" statement. In distributed systems, this object may be on one of the various processors and memories that constitute the distributed system. The time required for this access will involve both a communication channel delay and processing time on both processors involved. This delay will almost certainly be several orders of magnitude slower than accessing directly addressable objects.

Another serious disadvantage of distributing the AOTS prototype is the state-of-the art of procedures to distribute a program. Today, the computer science disciplines have not progressed to the point that Ada programs can be distributed with tools. The distribution task remains an art, rather than a science, for the software engineer.

B. Personal Computer Based Architecture.

This paragraph identifies the changes and the associated effort that is required to evolve the AOTS prototype software into the Personal Computer Based Concept described in Section I.C. The software architecture for a Personal Computer-based concept was previously shown in Figure B-2. Each AOTS unit was allocated to one of the Computer Program Configuration Items (CPCI) in Table B-7.

Based on these data, a number of changed lines is estimated from parallels drawn on the recent ISS experience in porting ISS to a microcomputer and the recent ISS analysis that evaluates the utility of ISS for the ATS program. This analysis ignores the long term cost of ownership which may very well determine whether the prototype software is ported or re-designed.

A set of requirements for the full scale development program was determined. This set of requirements then became the measure to gauge the utility of the existing software in the future architecture. The existing code was allocated to one of the CPCI if it provided functionality that could be used in the fabrication of that CPCI. Non-existent functions are assigned an estimated lines of code to implement. Prototype source lines of code are allocated to each of the CPCIs in Table B-7. Table B-7 is sorted by CPCI and lists each unit name or configuration number, source code language, and source lines of code (SLOC). A summary total lines and change lines is listed for the CPCI.

The criteria for adding each existing code unit to a CPCI are: the functionality of the code is required by the FSD AOTS; the Code is independent of the hardware; the comments are meaningful and complete, the code is written for the SQL interface to the DBMS and the POSIX interface to the operating system.

The discussion that follows assume that the prototype software is being ported to a single microcomputer. The code contains the same functionality and capability as the prototype. The code is written for the SQL and POSIX (or other standard operating system interface). Some comment lines are added to the code to improve understanding. The capability of the prototype is expanded to handle up to four hundred AFSs. The following numbers do not add archive and backup, or build supportability tools.

A basic requirement for the reuse or porting of the Prototype AOTS is the expansion of the prototype AFS-handling capability. It is considered unlikely that the next user of AOTS will be satisfied with exactly four AFSs, but some number reasonably will be desired by that future user. The Prototype MTL design is deficient for the small PC scenario. Currently, an MTL installation for four AFSs runs in the background in two to three hours. It is reasonable to assume that some nominal number of MTLs are installed at any given time. Perhaps a new MTL is installed for a given AFS every six months. In a simple math exercise, then, 400 AFSs would be installed in two to three hundred hours on the VAX 8650. This would be a semiannual requirement and all four hundred AFSs could be handled by a VAX- based system. Since the operation involves significant Input/Output (I/O) operations, and the PC processor speed is much slower than the VAX 8650, the prototype installation function is beyond the capability of a small PC.

The installation process for the prototype has not been optimized. No effort has been expended to streamline the MTL installation process. It is reasonable to expect that an investigation and re-design of the MTL files and installation processes would result in significant economies in file space and installation time. Another avenue is to perform the installations on a mainframe and then transfer the updated files to the PCs.

The discussion of the recent ISS analysis referred to in Section II.A.3. is largely applicable to this scenario. In that scenario, the code was allocated to Computer Program. The CPC terminology is consistent with the prototype development methodology, but for this and other future scenarios, the code is allocated to Computer Software Configuration Items to conform with more modern terminology.

The CPCIs for this scenario correspond very closely with the CPCs of the previous section. CPCIs are established for Management Application, Training Delivery, Training Development, Evaluation Application, Evaluation DEvelopment, Task Analysis Requirements Development, Virtual Machine Layer, System Support and General Support.

The System Support and General Support CPCIs may be significantly modified in order to make the program fit on a PC- sized machine. If the code is installed on a 80386-based or similar machine, the whole functionality and expanded AFS capability may fit with the exact same lines of code change discussed in Section II.A.3. for the simple porting case. On the other hand, if the code must be modified to fit on a smaller, less powerful machine, some things must be sacrificed. The system would become less automatic and the user would be required to perform more functions manually.

The AOTS prototype allows the multiple users to choose from a menu of Training Management, Task List Interfacing, Position Requirements, On-line Training, or System Utilities. Since the PC has limited active memory and disk storage, one machine may be dedicated to training delivery but would not have enough space for training management. Either specific machines would be dedicated to specific functions, or the machines would be reconfigured to change from one set of user functions to another. The user would assume more administrative and management burdens.

A simpler machine would not incorporate the complexity in System Support and General Support, and the virtual machine layer. Table B-4 is a summary of the above findings for the smaller PC machine. The findings for a larger PC is the same as Section I.E. findings.

Table B-4. SLOC Line Changes Required for PC Based Architecture.

CSCI NAME	SLOC	CHANGED LINES
Management Application	15,734	3,740
Evaluation Application	5,568	1,338
Training Delivery	46,709	10,510
Training Development	99,747	22,443
Evaluation Development	4,269	960
Task Analysis and Requirements Development	18,554	5,160
Virtual Machine Layer	0	0
System Support	1,534	345
General Support	3,604	3,342

Total SLOC:	195,720
Total Changed Lines:	47,840
Percent Change:	24%

C. Networked System Architecture

The changes required and the associated effort to convert the AOTS prototype software to the Networked System Architecture described in Section I.D. of this Appendix are estimated in this section.

As was done for the previous scenarios, a set of requirements for the full scale development program was decided upon. This set of requirements became the measure to gauge the utility of the existing software in the future form. These future requirements are discussed earlier in this Appendix.

After the set of future requirements was determined, the existing code was allocated to one of ten CPCIs, if it provided functionality that could be used in the fabrication of that CPCI. Non-existent functions are assigned an estimated lines of code to implement.

Prototype source lines of code are allocated to each of the 10 CPCIs in Table B-8. Table B-8 is sorted by CPCI and lists each unit configuration number or name, source code language, source lines of code (SLOC), and CPCI number.

The criteria for adding each existing code unit to a CPCI are: the functionality of the code is required by the FSD AOTS, the Code is hardware independent, the comments are meaningful and complete, and the code is written for the SQL interface to the DBMS and the POSIX interface to the operating system.

The code in each CPCI was examined to determine to what extent it meets the criteria and a number of changes lines was estimated. The estimates that follow make certain assumptions about the features of the source code. The prototype software is being ported to a networked group of microcomputers. The code contains the same functionality and capability as the prototype. The code is re-written to include the SQL and POSIX interfaces. Some comment lines are added to the code to improve understanding. Office tools have been integrated into the functionality.

The Local Application functions have been included to integrate office tools into the FSD AOTS environment. Local printing and editing can be done at the PC. The program does not rely on the operating system as the free standing PC did, and low level VML functions have been added back into the scenario. The General Support functions have been bolstered to increase the automation of the AOTS processes.

Screens would be designed to allow paging through the expanded list of AFSs instead of the single screen display that is satisfactory now for the four AFSs. Reports would be reformatted, but they reside in code that is tagged for a full re-design, therefore the new formats will not add to the total SLOC to be re-written. Total new lines for AFS expansion changes is estimated at 480 new and 486 re-designed lines. This analysis ignores the long term cost of ownership which may very well determine the cost benefit of porting the prototype software.

Table B-5. SLOC Changes Required for Networked System.

CSCI NAME	SLOC	CHANGED LINES
Local Applications	7,500	7,500
Management Application	15,734	3,740
Evaluation Application	5,568	1,338
Training Delivery	46,709	10,510
Training Development	99,747	22,443
Evaluation Development	4,269	961
Task Analysis & Requirements Development	18,554	5,161
Virtual Machine Layer	6,975	6,975
System Support	15,290	7,645
General Support	22,354	21,927

Total SLOC:	242,700
Total Changed Lines:	88,199
Percent Change:	36%

Table B-6. Port to a Mainframe

CPC	Package Number	Language	SLOC	Lines Changed
Management Application	MGT 23	Ada	414	
	MGT 23	Ada	105	
	MGT 30	Ada	1358	
	MGT 30	Ada	35	
	MGT 31	Ada	36	
	MGT 31	Ada	703	
	MGT 32	Ada	900	
	MGT 32	Ada	93	
	MGT 33	Ada	733	
	MGT 33	Ada	141	
	MGT 38	Ada	214	
	MGT 38	Ada	38	
	MGT 39	Ada	623	
	MGT 39	Ada	6	
	MGT 40	Ada	240	
	MGT 40	Ada	160	
	MGT 42	Ada	346	
	MGT 42	Ada	79	
	MGT 43	Ada	76	
	MGT 43	Ada	42	
	MGT 45	Ada	432	
	MGT 45	Ada	94	
	MGT 46	Ada	240	
	MGT 46	Ada	85	
	MGT 47	Ada	9	
	MGT 47	Ada	339	
	MGT 48	Ada	420	
	MGT 48	Ada	86	
	MGT 49	Ada	266	
	MGT 49	Ada	9	
	MGT 54	Ada	1926	
	MGT 54	Ada	77	
	MGT 62	Ada	258	
	MGT 62	Ada	89	
	MGT 63	Ada	512	
	MGT 64	Ada	219	
	MGT 64	Ada	72	
	MGT 65	Ada	450	
	MGT 65	Ada	473	
	MGT 86	Ada	198	
	MGT 86	Ada	13	
	MGT 87	Ada	519	

Table B-6. Port to a Mainframe (Continued)

CPC	Package Number	Language	SLOC	Lines Changed
	MGT 114	Ada	632	
	MGT 115	Ada	79	
	MGT 118	Ada	296	
	MGT 120	Ada	179	
Totals			15734	3740
Evaluation Application	EVL 8	Ada	908	
	EVL 8	Ada	123	
	EVL 9	Ada	457	
	EVL 9	Ada	75	
	EVL 16	Ada	309	
	EVL 16	Ada	73	
	EVL 17	Ada	787	
	EVL 17	Ada	92	
	EVL 18	Ada	794	
	EVL 18	Ada	108	
	EVL 19	Ada	639	
	EVL 19	Ada	133	
	EVL 20	Ada	23	
	EVL 21	Ada	346	
	EVL 21	Ada	86	
	EVL 22	Ada	243	
	EVL 22	Ada	14	
	EVL 23	Ada	269	
	EVL 24	Ada	84	
	MGT 28	Ada	5	
Totals			5568	1338
Training Delivery	CAILIB	Ada	3834	
	UTIL	Ada	14861	
	BGMON	Ada	2892	
	TIE	Ada	2544	
	TESTED	Ada	11422	
	CAIUTL	Ada	7377	
	CAIREP	Ada	3779	
Totals			46709	10510
Training Development	CASS	Ada	15750	
	EXTLIB	Ada	11475	
	GREDIT	Ada	11475	
	HLP	Ada	1275	
	MSG	Ada	1275	
	SE	Ada	25875	

Table B-6. Port to a Mainframe (Continued)

CPC	Package Number	Language	SLOC	Lines Changed
	CAGEN	Ada	406	
	CAGEN_	Ada	123	
	CTCR_	Ada	128	
	CTEXP_	Ada	112	
	GRLIB	Ada	915	
	GRLIB_	Ada	20	
	ITEM_SUP_	Ada	136	
	GRTYPES_	Ada	185	
	SDOKEYTYP	Ada	18	
	SD3SR_	Ada	18	
	SD4PASS_	Ada	51	
	SDINTSTO_	Ada	15	
	APPLIB	Ada	9000	
Totals			99747	22443
Evaluation Development				
	0	Ada	300	
	1	Ada	1647	
	1	Ada	190	
	2	Ada	198	
	2	Ada	20	
	3	Ada	57	
	5	Ada	590	
	5	Ada	17	
	6	Ada	220	
	6	Ada	116	
	7	Ada	648	
	7	Ada	45	
	12	Ada	138	
	12	Ada	83	
Totals			4269	961
Task Analysis	MGT 2	Ada	2388	
	MGT 2	Ada	110	
	MGT 3	Ada	1083	
	MGT 3	Ada	9	
	MGT 4	Ada	309	
	MGT 4	Ada	46	
	MGT 5	Ada	87	
	MGT 5	Ada	157	
	MGT 7	Ada	1749	
	MGT 7	Ada	231	
	MGT 9	Ada	64	

Table B-6. Port to a Mainframe (Continued)

CPC	Package Number	Language	SLOC	Lines Changed
MGT	19	Ada	4	
MGT	24	Ada	904	
MGT	24	Ada	87	
MGT	25	Ada	4	
MGT	50	Ada	390	
MGT	50	Ada	111	
MGT	51	Ada	107	
MGT	51	Ada	77	
MGT	52	Ada	33	
MGT	52	Ada	20	
MGT	55	Ada	1426	
MGT	55	Ada	138	
MGT	57	Ada	412	
MGT	58	Ada	31	
MGT	59	Ada	80	
MGT	60	Ada	1152	
MGT	60	Ada	222	
MGT	61	Ada	158	
MGT	66	Ada	27	
MGT	91	Ada	40	
MGT	92	Ada	533	
MGT	92	Ada	23	
MGT	93	Ada	866	
MGT	93	Ada	71	
MGT	94	Ada	576	
MGT	94	Ada	9	
MGT	95	Ada	251	
MGT	95	Ada	216	
MGT	96	Ada	120	
MGT	96	Ada	52	
MGT	97	Ada	87	
MGT	97	Ada	22	
MGT	98	Ada	135	
MGT	98	Ada	24	
MGT	99	Ada	64	
MGT	100	Ada	465	
MGT	100	Ada	141	
MGT	102	Ada	365	
MGT	102	Ada	100	
MGT	103	Ada	29	
MGT	103	Ada	53	
MGT	104	Ada	0	
MGT	105	Ada	0	

Table B-6. Port to Mainframe (Continued)

CPC	Package Number	Language	SLOC	Lines Changed
	SUP 28	Ada	69	
	SUP 28	Ada	87	
	SUP 29	Ada	131	
	SUP 29	Ada	91	
	SUP 30	Ada	84	
	SUP 31	Ada	337	
	SUP 31	Ada	88	
	SUP 32	Ada	73	
	SUP 32	Ada	16	
	SUP 33	Ada	508	
	SUP 34	Ada	11	
Totals			15290	7645
General Support	MGT 63	Ada	233	
	MGT 68	Macro	214	
	MGT 85	Ada	63	
	MGT 90	Ada	51	
	MGT XXX	SAS	3053	
Totals			3604	3343
Grand Total			216450	62115

Percent Changed Lines: 29%

Table B-7. Standalone Personal Computer

CPC	Package Number	Language	SLOC	Lines Changed
Management Application	MGT 23	Ada	414	
	MGT 23	Ada	105	
	MGT 30	Ada	1358	
	MGT 30	Ada	35	
	MGT 31	Ada	36	
	MGT 31	Ada	703	
	MGT 32	Ada	900	
	MGT 32	Ada	93	
	MGT 33	Ada	733	
	MGT 33	Ada	141	
	MGT 38	Ada	214	
	MGT 38	Ada	38	
	MGT 39	Ada	623	
	MGT 39	Ada	6	
	MGT 40	Ada	240	
	MGT 40	Ada	160	
	MGT 42	Ada	346	
	MGT 42	Ada	79	
	MGT 43	Ada	76	
	MGT 43	Ada	42	
	MGT 45	Ada	432	
	MGT 45	Ada	94	
	MGT 46	Ada	240	
	MGT 46	Ada	85	
	MGT 47	Ada	9	
	MGT 47	Ada	339	
	MGT 48	Ada	420	
	MGT 48	Ada	86	
	MGT 49	Ada	266	
	MGT 49	Ada	9	
	MGT 54	Ada	1926	
	MGT 54	Ada	77	
	MGT 62	Ada	258	
	MGT 62	Ada	89	
	MGT 63	Ada	512	
	MGT 64	Ada	219	
	MGT 64	Ada	72	
	MGT 65	Ada	450	
	MGT 65	Ada	473	
	MGT 86	Ada	198	
	MGT 86	Ada	13	
	MGT 87	Ada	519	
	MGT 87	Ada	66	

Table B-7. Standalone Personal Computer (Continued)

CPC	Package Number	Language	SLOC	Lines Changed
	MGT 114	Ada	632	
	MGT 115	Ada	79	
	MGT 118	Ada	296	
	MGT 120	Ada	179	
Totals			15734	3740
Evaluation Application	EVL 8	Ada	908	
	EVL 8	Ada	123	
	EVL 9	Ada	457	
	EVL 9	Ada	75	
	EVL 16	Ada	309	
	EVL 16	Ada	73	
	EVL 17	Ada	787	
	EVL 17	Ada	92	
	EVL 18	Ada	794	
	EVL 18	Ada	108	
	EVL 19	Ada	639	
	EVL 19	Ada	133	
	EVL 20	Ada	23	
	EVL 21	Ada	346	
	EVL 21	Ada	86	
	EVL 22	Ada	243	
	EVL 22	Ada	14	
	EVL 23	Ada	269	
	EVL 24	Ada	84	
	MGT 28	Ada	5	
Totals			5568	1338
Training Delivery	CAILIB	Ada	3834	
	UTIL	Ada	14861	
	BGMON	Ada	2892	
	TIE	Ada	2544	
	TESTED	Ada	11422	
	CAIUTL	Ada	7377	
	CAIREP	Ada	3779	
Totals			46709	10510
Training Development	CASS	Ada	15750	
	EXTLIB	Ada	11475	
	GREDIT	Ada	11475	
	HLP	Ada	1275	
	MSG	Ada	1275	
	SE	Ada	25875	

Table B-7. Standalone Personal Computer (Continued)

CPC	Package	Number	Language	SLOC	Lines Changed
	CTCR_	84	Ada	128	
	CTEXP	69	Ada	112	
	GRLIB	915	Ada	915	
	GRLIB	9	Ada	20	
	ITEM_	0	Ada	136	
	GRTYP	135	Ada	185	
	SDOKE	18	Ada	18	
	SD3SR	18	Ada	18	
	SD4PA	51	Ada	51	
	SDINT	8	Ada	15	
	APPLIB		Ada	9000	
Totals				99747	22443
Evaluation Development	EVL	0	Ada	300	
	EVL	1	Ada	1647	
	EVL	1	Ada	190	
	EVL	2	Ada	198	
	EVL	2	Ada	20	
	EVL	3	Ada	57	
	EVL	5	Ada	590	
	EVL	5	Ada	17	
	EVL	6	Ada	220	
	EVL	6	Ada	116	
	EVL	7	Ada	648	
	EVL	7	Ada	45	
	EVL	12	Ada	138	
	EVL	12	Ada	83	
Totals				4269	961
Task Analysis	MGT	2	Ada	2388	
	MGT	2	Ada	110	
	MGT	3	Ada	1083	
	MGT	3	Ada	9	
	MGT	4	Ada	309	
	MGT	4	Ada	46	
	MGT	5	Ada	87	
	MGT	5	Ada	157	
	MGT	7	Ada	1749	
	MGT	7	Ada	231	
	MGT	9	Ada	64	
	MGT	15	Ada	114	
	MGT	15	Ada	82	
	MGT	16	Ada	743	
	MGT	16	Ada	107	

Table B-7. Standalone Personal Computer (Continued)

CPC	Package Number	Language	SLOC	Lines Changed
MGT	51	Ada	107	
MGT	51	Ada	77	
MGT	52	Ada	33	
MGT	52	Ada	20	
MGT	55	Ada	1426	
MGT	55	Ada	138	
MGT	57	Ada	412	
MGT	58	Ada	31	
MGT	59	Ada	80	
MGT	60	Ada	1152	
MGT	60	Ada	222	
MGT	61	Ada	158	
MGT	66	Ada	27	
MGT	91	Ada	40	
MGT	92	Ada	533	
MGT	92	Ada	23	
MGT	93	Ada	866	
MGT	93	Ada	71	
MGT	94	Ada	576	
MGT	94	Ada	9	
MGT	95	Ada	251	
MGT	95	Ada	216	
MGT	96	Ada	120	
MGT	96	Ada	52	
MGT	97	Ada	87	
MGT	97	Ada	22	
MGT	98	Ada	135	
MGT	98	Ada	24	
MGT	99	Ada	64	
MGT	100	Ada	465	
MGT	100	Ada	141	
MGT	102	Ada	365	
MGT	102	Ada	100	
MGT	103	Ada	29	
MGT	103	Ada	53	
MGT	104	Ada	0	
MGT	105	Ada	0	
MGT	106	Ada	0	
MGT	107	Ada	0	
MGT	108	Ada	0	
MGT	109	Ada	0	
MGT	116	Ada	575	
MGT	116	Ada	24	

Table B-7. Standalone Personal Computer (Continued)

CPC	Package Number	Language	SLOC	Lines Changed
System Support	AO1US 23	Ada	12	
	AO1US 71	Ada	51	
	LGLIB 80	Ada	55	
	LGLIB 6	Ada	6	
	LGTYP 167	Ada	97	
	SACCE 37	Ada	23	
	SACCE 62	Ada	40	
	SUP 12	Ada	178	
	SUP 1	Ada	2748	
	SUP 1	Ada	451	
	SUP 4	Ada	108	
	SUP 4	Ada	71	
	SUP 5	Ada	88	
	SUP 5	Ada	861	
	SUP 6	Ada	97	
	SUP 6	Ada	56	
	SUP 7	Ada	394	
	SUP 7	Ada	131	
	SUP 9	Ada	0	
	SUP 9	Ada	16	
	SUP 10	Ada	781	
	SUP 11	Ada	879	
	SUP 12	Ada	850	
	SUP 13	Ada	59	
	SUP 13	Ada	30	
	SUP 14	Ada	692	
	SUP 14	Ada	100	
	SUP 15	Ada	233	
	SUP 15	Ada	147	
	SUP 16	Ada	25	
	SUP 16	Ada	171	
	SUP 17	Ada	887	
	SUP 17	Ada	257	
	SUP 18	Ada	0	
	SUP 19	Macro	727	
	SUP 20	Ada	92	
	SUP 21	Ada	203	
	SUP 21	Ada	203	
	SUP 22	Ada	105	
	SUP 22	Ada	128	
	SUP 23	Ada	77	
	SUP 23	Ada	99	
	SUP 24	Ada	78	

Table B-7. Standalone Personal Computer (Continued)

CPC	Package Number	Language	SLOC	Lines Changed
	SUP 30	Ada	84	
	SUP 31	Ada	337	
	SUP 31	Ada	88	
	SUP 32	Ada	73	
	SUP 32	Ada	16	
	SUP 33	Ada	508	
	SUP 34	Ada	11	
Totals			1535	345
General Support	MGT 63	Ada	223	
	MGT 68	Macro	214	
	MGT 85	Ada	63	
	MGT 90	Ada	51	
	MGT XXX	SAS	3053	
Totals			3604	3343
Grand Total			195720	47840

Percent Re-write: 24%

Table B-8. Networked Architecture.

CPC	Package Number	Language	SLOC	Lines Changed
Local Applications		Ada	1875	
		Ada	3750	
			1875	
Totals			7500	7500
Management Application	MGT 23	Ada	414	
	MGT 23	Ada	105	
	MGT 30	Ada	1358	
	MGT 30	Ada	35	
	MGT 31	Ada	36	
	MGT 31	Ada	703	
	MGT 32	Ada	900	
	MGT 32	Ada	93	
	MGT 33	Ada	733	
	MGT 33	Ada	141	
	MGT 38	Ada	214	
	MGT 38	Ada	38	
	MGT 39	Ada	623	
	MGT 39	Ada	6	
	MGT 40	Ada	240	
	MGT 40	Ada	160	
	MGT 42	Ada	346	
	MGT 42	Ada	79	
	MGT 43	Ada	76	
	MGT 43	Ada	42	
	MGT 45	Ada	432	
	MGT 45	Ada	94	
	MGT 46	Ada	240	
	MGT 46	Ada	85	
	MGT 47	Ada	9	
	MGT 47	Ada	339	
	MGT 48	Ada	420	
	MGT 48	Ada	86	
	MGT 49	Ada	266	
	MGT 49	Ada	9	
	MGT 54	Ada	1926	
	MGT 54	Ada	77	
	MGT 62	Ada	258	
	MGT 62	Ada	89	
	MGT 63	Ada	512	
	MGT 64	Ada	219	
	MGT 64	Ada	72	
	MGT 65	Ada	450	

Table B-8. Networked Architecture (Continued)

CPC	Package Number	Language	SLOC	Lines Changed
	MGT 114	Ada	632	
	MGT 115	Ada	79	
	MGT 118	Ada	296	
	MGT 120	Ada	179	
Totals			15734	3740
Evaluation Application	EVL 8	Ada	908	
	EVL 8	Ada	123	
	EVL 9	Ada	457	
	EVL 9	Ada	75	
	EVL 16	Ada	309	
	EVL 16	Ada	73	
	EVL 17	Ada	787	
	EVL 17	Ada	92	
	EVL 18	Ada	794	
	EVL 18	Ada	108	
	EVL 19	Ada	639	
	EVL 19	Ada	133	
	EVL 20	Ada	23	
	EVL 21	Ada	346	
	EVL 21	Ada	86	
	EVL 22	Ada	243	
	EVL 22	Ada	14	
	EVL 23	Ada	269	
	EVL 24	Ada	84	
	MGT 28	Ada	5	
Totals			5568	1338
Training Delivery	CAILIB	Ada	3834	
	UTIL	Ada	14861	
	BGMON	Ada	2892	
	TIE	Ada	2544	
	TESTED	Ada	11422	
	CAIUTL	Ada	7377	
	CAIREP	Ada	3779	
Totals			46709	10510
Training Development	CASS	Ada	15750	
	EXTLIB	Ada	11475	
	GREDIT	Ada	11475	
	HLP	Ada	1275	
	MSG	Ada	1275	

Table B-8. Networked Architecture (Continued)

CPC	Package Number	Language	SLOC	Lines Changed
	LGLIB 0	Ada	79	
	CACR 318	Ada	318	
	CACR_ 21	Ada	33	
	CAGEN 406	Ada	406	
	CAGEN 67	Ada	123	
	CTCR_ 84	Ada	128	
	CTEXP 69	Ada	112	
	GRLIB 915	Ada	915	
	GRLIB 9	Ada	20	
	ITEM_ 0	Ada	136	
	GRTYP 135	Ada	185	
	SDOKE 18	Ada	18	
	SD3SR 18	Ada	18	
	SD4PA 51	Ada	51	
	SDINT 8	Ada	15	
	APPLIB	Ada	9000	
Totals			99747	22443
Evaluation Development	EVL 0	Ada	300	
	EVL 1	Ada	1647	
	EVL 1	Ada	190	
	EVL 2	Ada	198	
	EVL 2	Ada	20	
	EVL 3	Ada	57	
	EVL 5	Ada	590	
	EVL 5	Ada	17	
	EVL 6	Ada	220	
	EVL 6	Ada	116	
	EVL 7	Ada	648	
	EVL 7	Ada	45	
	EVL 12	Ada	138	
	EVL 12	Ada	83	
Totals			4269	961
Task Analysis	MGT 2	Ada	2388	
	MGT 2	Ada	110	
	MGT 3	Ada	1083	
	MGT 3	Ada	9	
	MGT 4	Ada	309	
	MGT 4	Ada	46	
	MGT 5	Ada	87	
	MGT 5	Ada	157	
	MGT 7	Ada	1749	

Table B-8. Networked Architecture (Continued)

CPC	Package Number	Language	SLOC	Lines Changed
MGT	17	Ada	223	
MGT	17	Ada	44	
MGT	18	Ada	758	
MGT	18	Ada	26	
MGT	19	Ada	4	
MGT	24	Ada	904	
MGT	24	Ada	87	
MGT	25	Ada	4	
MGT	50	Ada	390	
MGT	50	Ada	111	
MGT	51	Ada	107	
MGT	51	Ada	77	
MGT	52	Ada	33	
MGT	52	Ada	20	
MGT	55	Ada	1426	
MGT	55	Ada	138	
MGT	57	Ada	412	
MGT	58	Ada	31	
MGT	59	Ada	80	
MGT	60	Ada	1152	
MGT	60	Ada	222	
MGT	61	Ada	158	
MGT	66	Ada	27	
MGT	91	Ada	40	
MGT	92	Ada	533	
MGT	92	Ada	23	
MGT	93	Ada	866	
MGT	93	Ada	71	
MGT	94	Ada	576	
MGT	94	Ada	9	
MGT	95	Ada	251	
MGT	95	Ada	216	
MGT	96	Ada	120	
MGT	96	Ada	52	
MGT	97	Ada	87	
MGT	97	Ada	22	
MGT	98	Ada	135	
MGT	98	Ada	24	
MGT	99	Ada	64	
MGT	100	Ada	465	
MGT	100	Ada	141	
MGT	102	Ada	365	
MGT	102	Ada	100	

Table B-8. Networked Architecture (Continued)

CPC	Package Number	Language	SLOC	Lines Changed
	MGT 110	Ada	0	
	MGT 111	Ada	0	
	MGT 116	Ada	575	
	MGT 116	Ada	24	
	MGT 117	Ada	0	
Totals			18554	5161
Virtual Machine Layer		Macro	1875	
		FORTRAN	5100	
Totals			6975	6975
System Support	AO1US 23	Ada	12	
	AO1US 71	Ada	51	
	LGLIB 80	Ada	55	
	LGLIB 6	Ada	6	
	LGTYP 167	Ada	97	
	SACCE 37	Ada	23	
	SACCE 62	Ada	40	
	SUP 12	Ada	178	
	SUP 1	Ada	2748	
	SUP 1	Ada	451	
	SUP 4	Ada	108	
	SUP 4	Ada	71	
	SUP 5	Ada	88	
	SUP 5	Ada	861	
	SUP 6	Ada	97	
	SUP 6	Ada	56	
	SUP 7	Ada	394	
	SUP 7	Ada	131	
	SUP 9	Ada	0	
	SUP 9	Ada	16	
	SUP 10	Ada	781	
	SUP 11	Ada	879	
	SUP 12	Ada	850	
	SUP 13	Ada	59	
	SUP 13	Ada	30	
	SUP 14	Ada	692	
	SUP 14	Ada	100	
	SUP 15	Ada	174	
	SUP 15	Ada	147	
	SUP 16	Ada	25	
	SUP 16	Ada	171	
	SUP 17	Ada	887	

Table B-8. Networked Architecture (Continued)

CPC	Package Number	Language	SLOC	Lines Changed
	SUP 23	Ada	77	
	SUP 23	Ada	99	
	SUP 24	Ada	78	
	SUP 24	Ada	15	
	SUP 26	Ada	715	
	SUP 26	Ada	138	
	SUP 27	Ada	683	
	SUP 28	Ada	69	
	SUP 28	Ada	87	
	SUP 29	Ada	131	
	SUP 29	Ada	91	
	SUP 30	Ada	84	
	SUP 31	Ada	337	
	SUP 31	Ada	88	
	SUP 32	Ada	73	
	SUP 32	Ada	16	
	SUP 33	Ada	508	
	SUP 34	Ada	11	
Totals			15290	7645
General Support		Ada	7500	
		Ada	3750	
		Ada	7500	
	MGT 63	Ada	223	
	MGT 68	Macro	214	
	MGT 85	Ada	63	
	MGT 90	Ada	51	
	MGT XXX	SAS	3053	
Totals			22354	21927
Grand Total			242700	88199

Percent Change Line: 36%

APPENDIX C ACRONYMS

ACS	VAX Ada Code Library System
ADS	Automated Data System
AF	Air Force
AFB	Air Force Base
AFCC	Air Force Communications Command
AFCMD	Air Force Contract Management Division
AFCSA	Air Force Communication-Computer Systems Architecture
AFHRL	Air Force Human Resources Laboratory
AFISA	Air Force Information System Architecture
AFLC	Air Force Logistics Command
AFM	Air Force Manual
AFMPC	Air Force Military Personnel Center
AFOMC	Air Force Occupational Measurement Center
AFR	Air Force Regulation
AFRES	Air Force Reserves
AFS	Air Force Specialty
AFSC	Air Force Specialty Code
AFSC	Air Force Systems Command
AFSS	Air Force Security Service
ALC	Air Logistics Center
ALCCM	AOTS Life Cycle Cost Model
ALPS 2000	Model 2000, ALPS brand name dot matrix printer
ANG	Air National Guard
ANGB	Air National Guard Base
ANSI	American National Standards Institute
AOTS	Advanced On-the-Job Training System
APDS	Automated Personnel Data System
APDS II	Next generation APDS
Assembler	Machine-level programming language
ATC	Air Training Command
ATR	Airman Training Records
ATS	Advanced Training System
AT&T 3B2	American Telephone and Telegraph mini-computer
Baud	Bit word rate of data transfer
BDM	Braddock, Dunn and MacDonald Corporation
Blue Suit	Air Force
BSED	Ball Systems Engineering Division
C-CS	Communication-Computer System
C-5	Computer software system specification
CAI	Computer Aided Instruction
CAMS	Core Automated Maintenance System
CBPO	Consolidated Base Personnel Office
CBT	Computer-Based Training
CDC	Career Development Course
CDRL	Contract Deliverable Requirements List
CER	Cost Estimating Relationship

CI	Configuration Item
CM	Configuration Management
CMP	Configuration Management Plan
CMS	Configuration Management System
Coaxial Cable	Shielded data transmission cable
COCOMO	COnstructive COst MOdel
Compiler	Software application program that converts source code into machine language
COTS	Commercial Off-The-Shelf
Courseware	Resources necessary to complete a specific training courses
CPC	Computer Program Component
CPCI	Computer Program Configuration Item
CRLCMP	Computer Resources Life Cycle Management Plan
CSAS	Configuration Status Accounting System
CSC	Computer Software Component
CSCI	Computer Software Configuration Item
CWBS	Cost Work Breakdown Structures
DAC	Douglas Aircraft Company
dba	Doing Business As
DBMS	Database Management System
DCA	Defense Communications Agency
DCL	DIGITAL Control Language
DDN	Defense Data Network
DEERS	Defense Enrollment Eligibility Reporting System
Disk Drive	Internal or external computer diskette reading device
DMS	Data Management System
DoD	Department of Defense
DoD-STD	Department of Defense Standard
DMS	Data Management System
DT&E	Development, Test and Evaluation
ECD	Estimated Completion Date
EGA	Enhanced Graphics Adapter
Expansion	Add on memory increasing the basic memory capacity of a computer's memory
Fiber Optics	Cable which allows for the use of light as a medium of communication
File Server	Large scale data storage device that is implemented on a LAN
Floppy Disk	Transportable computer storage medium
FORTRAN	FORmula TRANslation programming language
FMI	Functional Management Inspection
FSD	Full Scale Development
FTD	Field Training Detachment
FY	Fiscal Year
Gigabytes	One billion bytes
Global Data	Data that is accessible through all modules of an application

Global variable	Memory variable that is accessible through all modules of an application
GOSIP	Government Open Systems Interconnect Protocol
GPTR	Generic Position Task Requirements
GSU	Geographically Separated Unit
Hard Coded	Code that does not accommodate future changes
Hard Disk	Fixed computer storage medium
Hardware	Computer equipment
HRL/OL-AK	Human Resources Laboratory Operating Location AK
HSD	Human Services Division
IEEE	Institute of Electronics and Electrical Engineers
I&SM	Integration and Site Management
IG	Inspector General
ilities	Support requirements (i.e., reliability, maintainability, interoperability, etc.)
IOT&E	Initial Operational Test and Evaluation
IP	Internet Protocol
ISD	Information Systems Directive
ISD	Instructional System Development
ISO	International Standards Organization
ISS	Instructional Support System
ITR	Individual Training Record
IVD	Interactive Video Disk
JQS	Job Qualification Standard
K	1,000
Kernel	A self contained computing environment that allows an applications program to interact with a specific machine
Kilobaud	1,000 baud, rate of data transmission
LAN	Local Area Network
Life Cycle	Entire life of a project from initial design through operations and maintenance
Line Printer	Printer that prints one line at a time
LITA	Local Information Transfer Architecture
MAC	Military Air Command
Mainframe	Single, large central computer
MAISRC	Major Automated Information Systems Review Council
Magnetic Tape	An electronic storage medium
MAJCOM	Major Command
Mb	Mega byte
Mega	1 million
Microprocessor	Personal computer
MIL-STD	Military Standard
Mini Computer	Middle sized computer system
Modem	Device for computers to communicate over telephone lines
MPR	Military Personnel Record
MTL	Master Task List

Networked	Hardware/software that will allow computers to interface
OCD	Operational Concept Document
OJT	On-the-Job Training
OL	Operating Location
O&M	Operations and Maintenance
OMC	Occupational Measurement Center
OMR	Occupational Measurement Report
OPR	Office of Primary Responsibility
Optical Character Reader	Scanning device that reads characters
OPTR	Operational Position Training Requirements
Organic Support	"In-house" support group
OS	Operating System
OSI	Open System Interconnect
OTR	Other Training Requirements
Paybacks	Return on investment
PC III	Personnel Concepts III
PC	Personal Computer
PCS	Permanent Change of Station
PE	Program Element
PMD	Program Management Document
PME	Professional Military Education
PMP	Program Management Plan
POM	Program Objective Memorandum
Port	Transfer of a computer applications program between devices
POSIX	Portable Operating System Interface for computing environments
PPBS	Planning, Programming, and Budgeting System
P3I	Preplanned Product Improvements
QC	Quality Control
RAM	Random Access Memory
RCA Price S	RCA Price Systems cost estimation system
RDT&E	Research, Development, Test and Evaluation
REVIC	AFSC Contract Management Division version of COCOMO
R&D	Research and Development
RFP	Request for Proposal
RGB	Red, Green, Blue Color Monitor
SAC	Strategic Air Command
SAS	Statistical Analysis System
Shielded Broadband	Allows for a broad frequency spectrum to be passed through a shielded medium
Shift Fraction	Maximum number of people from an AFS which will be on duty at any one time
SLOC	Source Lines of Code
SLT&E	System Level Test and Evaluation

SM	Service Member
SME	Subject Matter Expert
Software	High and low level computer instructions
SORD	System Operational Requirements Document
Source Code	High level computer instructions
SOW	Statement of Work
SPAS	Security Police Automated System
SPO	System Program Office
SPR	Software Problem Report
SQL	Structured Query Language
SSAN	Social Security Account Number
SSC	Software Support Center
Standalone	Computer system that runs entirely on the PC microprocessor and supports single users
STD	Standard
Strawman	Initial concept document
STS	Specialty Training Standard
TAC	Tactical Air Command
Tapes	See magnetic tapes
TBD	To Be Determined
TCP	Transmission Control Protocol
TDY	Temporary Duty
Throughput	A measure of the speed of a computer
TMTL	Tentative Master Task List
TTC	Technical Training Center
Twisted Pair	Twisted copper cables normally used for telephone lines, can be more than two wires in a bundle
ULANA	Unified Local Area Network Architecture
UNISYS	UNIted SYStems Inc., manufacturer of Sperry computer systems
UNIX	Mini-computer operating system developed by AT&T
USAF	United States Air Force
USAF/DP	Air Staff Deputy for Personnel
USAFOMC	USAF Occupational Measurement Center
USDOT	US Department of Transportation
UTS	Universal Terminal System
VAX 8650	Digital Equipment Corporation mainframe computer
Verac	Original contractor, now dba BSED
VML	Virtual Machine Layer
VMS	Virtual Memory Operating System
V1	Version 1
Work center	Area where a group of people perform tasks
Work station	Area where independent or networked computer equipment is located
WR-ALC	Warner Robbins Air Logistics Center
WWMCCS	World Wide Military Command and Control System
Z-248	Zenith model 248 personal computer

APPENDIX D AOTS DOCUMENTATION

NO.	TITLE	SOURCE	DATE	STATUS
1	AOTS Life Cycle Cost Model Description	BSED	01/14/87	Complete
2	AOTS Life Cycle Final Report	BSED	05/30/87	Complete
3	CAMS Interface Study	BSED	06/22/87	Complete
4	EANGB Facilities Req Plan	BSED	08/31/87	Complete
5	Develop Baseline Test Design for Data Acquisition and Analysis	BSED	07/10/87	Complete
6	Recommendations for Major Drivers in the Sensitivity Analysis of the Cost Model Elements	BSED	10/01/87	Complete
7	SLT&E Readiness Planning			
	1.1 CDRL 6	BSED	03/23/88	Complete
	1.2 CDRL 3, 4	BSED	07/15/88	Complete
	1.3 CDRL 3, 4	BSED	08/11/88	Complete
8	Transition Plan	BSED	05/01/89	Complete
9	Perform Intermediate Analysis of SLT&E Data	BSED	(05/05/89)	ECD
10	AOTS Operational Concept Document	BSED	02/13/89	Complete
11	Expansion Plan	BSED	(06/01/89)	ECD
12	Final Technical Report	BSED	(07/31/89)	ECD

NO.	TITLE	SOURCE	DATE	STATUS
13	Management Subsystem Prime Item Spec	DAC	(02/03/86)	ECD
14	Training Dev and Delivery Subsystem Prime	DAC	02/03/86	Complete
15	Computer Support Subsystem Prime Item Specs	DAC	02/03/86	Complete
16	Hardware Component Critical Item Specs	DAC	02/03/86	Complete
17	Personnel and Support Subsystem Prime Item Spec	DAC	02/03/86	Complete
18	Management CPCI Development Spec	DAC	02/03/86	Being Revised
19	Evaluation CPCI Development Spec	DAC	02/03/86	Complete
20	Software Development Plan	DAC	02/03/86	Complete
21	Training Development and Delivery CPCI Development Spec	DAC	02/24/86	Complete
22	System Support CPCI Development Spec	DAC	02/24/86	Complete
23	Evaluation Subsystem Prime Item Spec	DAC	03/18/86	Complete
24	Computer Program Development Spec	DAC	03/20/86	Complete
25	Training Development and Delivery Subsystem Prime Item Spec	DAC	02/20/86	Complete
26	Trade Study Matrix Criteria CPU	DAC	03/25/86	Complete

NO.	TITLE	SOURCE	DATE	STATUS
27	Trade Study Matrix Criteria Terminal	DAC	03/25/86	Complete
28	Trade Study Matrix Criteria Printer/Plotter	DAC	03/25/86	Complete
29	Trade Study Matrix Criteria Ada Compiler and Support Draft of Plan for Ada Trade Study	DAC	03/25/86	Complete
30	Prime Item Dev Spec for Management Subsystem	DAC	03/28/86	Being Revised
31	Prime Item Dev Spec for Evaluation Subsystem	DAC	03/28/86	Being Revised
32	Prime Item Dev Spec for Personnel Subsystem	DAC	03/28/86	Being Revised
33	System Spec for the Training System	DAC	03/31/86	Being Revised
34	System Support Computer Program Configuration	DAC	04/17/86	Being Revised
35	Training Dev and Delivery Computer Program Config Item Dev Spec	DAC	04/17/86	Complete
36	Eval Computer Program Config Item Dev Spec	DAC	04/17/86	Complete
37	System Engineering Management Plan	DAC	09/30/86	Complete
38	System Specification	DAC	05/05/86	Being Revised

NO.	TITLE	SOURCE	DATE	STATUS
39	AOTS Phase II Hardware and Systems Engineering	DAC	05/23/86	Complete
40	Computer Programming Standard	DAC	09/16/88	Complete
41	Maintainability Program	DAC	09/12/86	Complete
42	Reliability Program Plan	DAC	09/10/86	Complete
43	Human Engineering Program Plan	DAC	01/29/87	Complete
44	Dev Spec Part II Eval, Com P, Config Item	DAC	11/14/86	Complete
45	Dev Spec Part II Sys Sup, Com Pro, Config Item	DAC	11/24/86	Complete
46	Maintenance Plan	DAC	10/30/86	Complete
47	Config Management Plan	DAC	12/18/86	Complete
48	Prime Item Dev Spec	DAC	03/11/86	Complete
49	Master Test Plan Part II	DAC	09/01/87	Complete
50	BAFB Site Prep Require and Install Plan	DAC	12/14/87	Complete
51	Update Maintenance Plan	DAC	06/03/88	Complete
52	Operational Guide w/wo Appendix B	DAC	06/10/88	Complete
53	Evaluation C5 Spec	DAC	08/17/88	Complete
54	Master Test Plan Part V	DAC	09/11/88	Complete
55	Master Test Plan Part IV	DAC	10/11/88	Complete

NO.	TITLE	SOURCE	DATE	STATUS
56	Eval Computer Program Config Spec Part II	DAC	11/03/88	Complete
57	System Support Computer Program Config Spec Part II	DAC	11/03/88	Complete
58	Management Computer Program Config Spec Part II	DAC	11/03/88	Complete
59	Volume I Users Handbook	DAC	12/05/88	Being Revised
60	Volume II Users Handbook	DAC	12/20/88	Being Revised
61	Management Subsystem Prime Item Spec	DAC	01/11/89	Complete
62	Volume III Users Handbook	DAC	01/11/89	Draft
63	Volume IV Users Handbook	DAC	01/17/89	Draft
64	System Spec for AOTS (2nd Draft)	DAC	01/20/89	Complete
65	AOTS Off-line Test Material Accountability Procedures	DAC	03/06/89	Complete
66	Reliability and Maintainability Program Report Draft	DAC	12/12/87	Complete
67	Technical Report	DAC	(10/31/89)	ECD
68	Functional Description of the AOTS (Section C, Appendix A)	DAC	09/30/83	Complete
69	IVD Instructional Plan	UES	07/88	Complete
70	IVD Story Boards	UES	08/88	Complete

NO.	TITLE	SOURCE	DATE	STATUS
71	Interactive Graphics	OAO	07/87	Complete
72	Static Graphics	OAO	09/87	Complete
73	Graphic Simulation	OAO	10/87	Complete
74	Engine Graphics	OAO	12/87	Complete
75	Graphic Development	OAO	11/88	Complete
76	Graphic Development	OAO		Working
77	Interim Technical Rpt	DAC	12/30/86	Complete
78	AOTS Life Cycle Cost Report (Blue Book)	BSED/AF	(07/01/89)	ECD
79	Procedures for Storing and Distributing AOTS Test Material Offline	DAC	04/12/89	Complete
80	Operational Guide	DAC	05/89	Complete
81	AOTS Offline Evaluation Materials Accountability Procedures	DAC	03/03/89	Complete
82	Design for Generating Quality Control Evaluation Events	DAC	03/89	Complete
83	Hardware Critical Item Spec	DAC	02/03/86	Complete
84	Master Test Plan	DAC	10/10/88	Complete
85	Advanced On-The-Job Training System (AOTS) Software Supportability Evaluation and Risk Assessment	BSED	12/15/88	Complete
86	Programmers Software Support Utilities (including AOTS version 2.4 tapes)	DAC	07/17/89	ECD

APPENDIX E REFERENCES

1. Advanced On-The-Job Training System (AOTS) Operational Concept Document, CDRL #13, 13 February 1989.

2. AOTS Life Cycle Cost Model Description, 14 January 1987.

3. Critical Item Development Specification for the Hardware Component of the AOTS, AFHRL Specification Number 70S64701, 3 February 1986.

4. Advanced On-The-Job Training System (AOTS) Software Supportability Evaluation and Risk Assessment, 15 December 1988.

5. AOTS Site Preparation Requirements and Equipment Installation Plan, DID L-5083, 26 January 1988.

6. Facilities Requirements Plan for the Transition of the AOTS to Ellington ANGB Tx, 16 October 1987.

7. AOTS System Specification, CI# 7000, 5 May 1986.

8. Management Subsystem Prime Item Specification, CI# 7100, 28 March 1986.

9. Training Development and Delivery Prime Item Specification, CI# 7200, 17 April 1986.

10. Evaluation Subsystem Prime Item Specification, CI #7300, 28 March 1986.

11. Computer Support Subsystem Prime Item Specification, CI# 7400, 3 February 1986.

12. Personnel and Logistic Support System Prime Item Specification, CI# 7500, 3 February 1986.

13. Management Computer Program Configuration Item Development Specification CPCI, CI# 7411, 3 February 1986.

14. Training Development and Delivery Subsystem Computer Program Configuration Item Development Specification CPCI, CI# 7412, 2 February 1986.

15. Evaluation Computer Program Configuration Item Development Specification CPCI, CI# 7413, 3 February 1986.

16. Evaluation Computer Program Configuration Item Product Specifications CPCI, CI# 7413 Volume II, 3 February 1986.

17. System Support Computer Program Configuration Item Development Specification CPCI, CI# 7414, 24 February 1986.

18. System Support Computer Program Configuration Item Product Specifications CPCI, CI# 7414 Volume II, 3 February 1986.

19. Management Computer Program Configuration Item Product Specifications CPCI, CI# 7411 Volume II, 3 February 1986.

20. Computer Standards Programming Document for Advanced On-the-Job Training System, 15 September 1986.

21. Advanced On-the-Job Training System Software Development Plan, 07 April 1986.

22. Master Test Plan, 10 October 1988.